VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA KYBERNETIKY A BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

BAKALÁŘSKÁ PRÁCE

2016 Jan Ručka

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA KYBERNETIKY A BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Automatizace měření laboratorní úlohy předání tepla

Automated Measurement System for Laboratory Experiment

2016 Jan Ručka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Jan Ručka**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: Automatizace měření laboratorní úlohy předání tepla
Automated Measurement System for Laboratory Experiment

Jazyk vypracování: čeština

Zásady pro vypracování:

Při sériové výrobě a v laboratořích se pro vyloučení lidského faktoru a pro urychlení měření a testů používají automatizované měřicí a testovací systémy. Práce se zabývá návrhem sběru dat a automatizace měření laboratorní úlohy a následně implementací.

Body zadání:
1. Seznamení se s problematikou automatizovaného měření a testování.
2. Seznámení se s principem laboratorní úlohy a s vlastnostmi senzorů.
3. Definice požadavků a návrh koncepce aplikace.
4. Implementace SW pro automatizaci měření.
5. Ověření funkčnosti aplikace, zhodnocení použité technologie, shrnutí výsledků.

Seznam doporučené odborné literatury:

[1] VLACH, Jaroslav, Viktorie VLACHOVÁ, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW.* 1. vyd. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
[2] BRESS, Thomas J. *Effective labview programming.* 1st ed. Allendale: NTS Press, 2013, 720 s. ISBN 1-934891-08-8/978-1-934891-08-7.
[3] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. *LabView advanced programming techniques.* 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 0-8493-3325-3.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 01.09.2015
Datum odevzdání: 29.04.2016

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * OSTRAVA * Fakulta elektrotechniky a informatiky

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

„Prohlašuj., že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal."

V Ostravě dne 26. 4. 2016 Podpis 

Poděkování

Rád bych poděkoval svému vedoucímu práce panu doc. Ing. Petru Bilíkovi, Ph.D. za vedení práce, důslednou kontrolu průběhu prací a také za pomoc při programování v prostředí LabVIEW. Dále bych chtěl poděkovat panu Ing. Davidu Valovi za pomoc v prvotních fázích při zjišťování komunikačních možností komponentů. Mé poděkování patří samozřejmě i přítelkyni a rodině, která mě podporovala a poskytovala částečnou kontrolu stylistiky práce.

Abstrakt

Tato práce obsahuje návrh a implementaci softwarové aplikace pro měření výměny tepla na již existujícím HW laboratorním modelu. Pro tuto práci byl zvolen postup skládající se ze tří částí. V první části byly zjištěny parametry užitých přístrojů a parametry programování komunikace založené na protokolu Modbus a rozhraní IO-Link. V druhé části proběhlo elektrické zapojení úlohy a v poslední části proběhl vývoj uživatelského rozhraní měřící aplikace v programovacím prostředí LabVIEW.

Konečným produktem této práce je program schopný kontinuálně měřit, vyhodnocovat a zapisovat do souboru procesní údaje. Program je schopen zobrazit aktuální hodnoty, graf hodnot s historií a uložit naměřené hodnoty do souboru.

Výsledky této práce rozšiřují množství reálných pracovních stanovišť pro studenty oboru Řídící a informační systémy, na kterých lze získávat zkušenosti z oblasti měřící a regulační techniky.

Klíčová slova

Modbus/TCP, LabVIEW, IO-Link, výměna tepla, měření, logování

Abstract

This bachelor thesis focuses on creating SW application for measuring Heat exchange on existing platform. This thesis chooses progress, which is divided into three parts. In first part the parameters of used devices and communication parameters via Modbus protocol and IO-Link interface were found out. In second part the process was mechanically connected. In third part an user-interface development for measurement in programming environment called LabVIEW was done.

Final product of this thesis is a program, which is able to measure, evaluate and log data continously. Evaluation involves in displaying actual measured data, displaying graph, which is able to show amount of previous measured data, and saving those data into the selected file.

This thesis results extend the amount of real working stations for students studying Control and Information Systems, on which man can gain experiences from Measurement and Control area.

Key words

Modbus/TCP , LabVIEW, IO-Link, Heat Exchange, Measurement, Logging

Obsah

Seznam použitých symbolů a zkratek

Zkratka	Význam
BOOL	Datový typ obsahující dvoustavovou hodnotu True nebo False zabírající adresový prostor 1 bitu
CAN	(Controller Area Network) Sériová sběrnice používána v automobilovém průmyslu
EtherNet/IP	Průmyslový komunikační protokol založený na Ethernetu, vhodný pro komunikaci převodníkem rozhraní s PLC přes sběrnici
HW	(Hardware) – fyzicky dostupné a realizovatelné zařízení
hex	Přípona hex označuje číslo zapsané v šestnáctkové soustavě
IEC	(International Electrotechnical Commission) Organizace, která vytváří mezinárodní normy pro elektrotechniku, elektroniku apod.
IODD	(IO Device Description) Soubor obsahující údaje o konkrétním zařízení (kódování dat, nastavení, apod.)
ISDU	(Indexed Service Data Unit) Indexované parametry zařízení
ISO/OSI	Vrstvový model vytvořený za účelem standardizace počítačových sítí
Modbus – IDA	Organizace vzniklá spojením Modbus organization a IDA Group
PC	Osobní počítač
PDU	(Protocol Data Unit) Zpráva na 6. úrovni ISO/OSI modelu
PLC	(Programmable Logic Controller) Programovatelný logický automat
ρ	Fyzikální veličina označující hustotu (kg/m^3)
SW	(Software) – programy vytvořené počítačem ovládající Hardware
TCP/IP	(Transmission Control Protocol/Internet Protocol) Rodina protokolů pro komunikaci v počítačové síti
UI	(User interface) Uživatelské rozhraní
UINT	Datový typ obsahující rozmezí hodnot od 0 do 65535 zabírající adresový prostor šestnácti bitů neboli dvou Bytů
WirelessHART	Bezdrátová síť založena na protokolu HART

Seznam použitých tabulek

Seznam použitých obrázků

Úvod

V této práci řeším automatizaci měření laboratorní úlohy výměny tepla kapalného média ve výměníku. Výměna tepla je v průmyslu často řešeným tématem, neboť vedlejším produktem přeměn různých energií je nežádoucí teplo. Toto teplo je nutno z procesu odebrat a právě k tomuto se využívá výměníků tepla. Základní charakteristiky použitého výměníku a jeho porovnání s jinými typy výměníků je popsáno v teoretické kapitole Výměníky tepla.

Tuto práci jsem si vybral, protože bych se rád ve svém pozdějším studiu a také práci zaměřil na návrh, realizaci a servis měřící a regulační techniky v průmyslu. Další motivací byl výběr ryze praktického tématu. Tyto témata jsou velice důležité pro následující výběr povolání, protože u nich si studenti mohou ujasnit, co by chtěli v budoucnu opravdu dělat.

Automatizace měření spočívá v propojení senzorů s počítačem pro usnadnění zjišťování naměřených hodnot a umožnění logování těchto hodnot.

Prvním úkolem pro zajištění automatizace byla identifikace procesního pracoviště. Této problematice se věnuje kapitola Popis laboratorní úlohy, podrobnějšímu popisu dostupných senzorů se věnuje kapitola Dostupný Hardware.

Druhý úkol představoval zajištění komunikace mezi senzory a počítačem. Pro komunikaci se senzory využívám rozhraní IO-Link zajišťující komunikaci na úrovni senzorů. Nadřazenou komunikaci s počítačem zajišťuje komunikační protokol Modbus. Veškeré prvky pro komunikaci jsou popsány v kapitole Komunikační prostředky.

Třetím a hlavním úkolem této práce je vlastní vývoj aplikací. Pro vývoj aplikací jsem vybral programovací prostředí LabVIEW, jehož teoretickému popisu se věnuje stejnojmenná kapitola. Praktickému vývoji aplikací se věnují kapitoly Návrh řešení a Realizace práce. Montážní práce na pracovišti jsou popsány v kapitole Mechanické práce a parametry komunikace.

V průběhu pracování jsou vytvořeny tři koncepty se vzrůstající složitostí, tak jak se zlepšovala má znalost použitých komponent a zvoleného programovacího prostředí. Další motivací pro neustálé vylepšování realizací je má věčná nespokojenost s momentálním řešením. Při vypracování jsem se řídil svým heslem: „Vždy je co zlepšovat“.

Mé hlavní cíle jsou tedy mechanická příprava pracoviště, tvorba aplikace pro vyhodnocení měření na pracovišti a také tvorba odpovídajícího uživatelského rozhraní této aplikace. Další vytyčené cíle jsou zdokumentování aplikace, vytvoření nápovědy pro běžné uživatele a zabezpečený nastavovací dialog. Zde bude možné nastavit základní parametry aplikace a všech přidružených částí. Aplikace bude především sloužit pro potřeby měření v laboratoři a také pro demonstrační účely.

Jiná řešení problematiky propojení IO-Link, Modbus protokolu a LabVIEW na úrovni bakalářské či diplomové práce se mi nepodařilo najít, což mírně zvýšilo obtížnost práce.

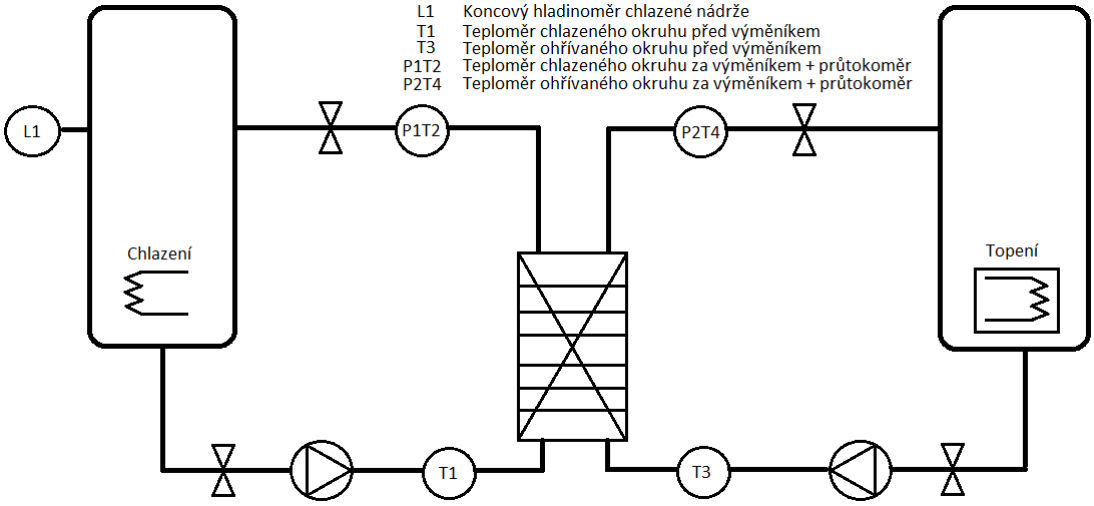
1 Popis laboratorní úlohy

Laboratorní model předání tepla po stránce mechanické a stránce senzorů se již na pracovišti nacházel. Scházela ovšem automatizace měření, která je hlavní náplní popisované práce a také elektrické zapojení senzorů.

Obrázek 1.1: Fotografie pracoviště, popisky na Obrázek 1.2

Pracoviště je možné rozdělit do dvou procesních okruhů propojených výměníkem tepla (Obrázek 1.1 a Obrázek 1.2). Levý okruh má nádrž chlazenou chladičem, který se běžně využívá pro chlazení nápojů. Pravý okruh má v nádrži topnou spirálu o výkonu 2 kW ohřívající teplonosné médium. Oba okruhy se dále sestávají z čerpadla, teploměru TN2531 (T1, T3), průtokoměru SM9000 (P1T2, P2T4) a dvou ventilů.

Teplonosná média proudící oběma okruhy si předávají teplo v pájeném deskovém výměníku ve spodní části pracoviště a vrací se do svých nádrží. Nádrže mají kapacitu přibližně 100 litrů, nicméně je zaplněna přibližně do dvou třetin.



Obrázek 1.2: Schéma pracoviště včetně vyznačených komponentů

Senzory, popsané v kapitole dostupný HW, se na pracovišti využívají následovně:

- Hladinoměr LMT121 (str. 25) hlídá minimální hladinu v nádrži s chlazeným teplonosným médiem (Hladinoměr L1)
- Teploměr TN2531 (str. 25Obrázek 4.2) měří teplotu teplonosného média na vstupech do výměníku obou procesních okruhů (Teploměr T1, T3)
- Průtokoměr SM9000 (str. 26) měřící aktuální průtočné množství (Průtokoměr P1, P2) a také teplotu na výstupu z výměníku obou procesních okruhů. (Teploměr T2, T4)

Popisovaná soustava výměny tepla má, z pohledu regulačního popisu, dvě akční veličiny (chod chladiče, chod topné spirály) a více regulovatelných parametrů, jako například teploty v nádržích, množství předaného tepla mezi okruhy, rychlost a kvalita regulace apod. Nicméně návrh regulace není součástí této práce.

Úkolem této bakalářské práce je navrhnout a realizovat automatizované měření a vyhodnocení procesních veličin.

2 Výměníky tepla

Výměníky tepla jsou zařízení pro uskutečnění průběžného nebo přerušovaného přenosu tepelné energie mezi dvěma, nebo více teplonosnými médii [1]. Výměníky tepla jsou nutné všude tam, kde je zapotřebí ochladit nebo ohřát procesní médium. V praxi se procesní média nutné k ochlazení nebo ohřevu omezují převážně na vodu, vodní páru, vzduch nebo spaliny, právě z důvodu jejich tepelných vlastností.

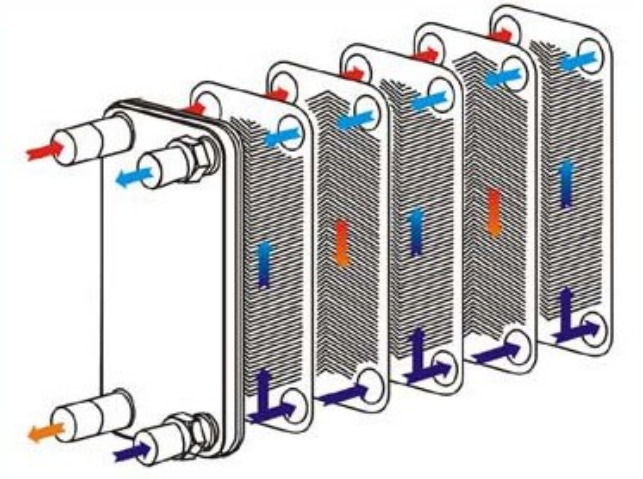
Výměníky tepla se dělí podle různých hledisek, z pohledu práce bude ale zmíněno pouze hledisko konstrukční. Dle konstrukčního hlediska se výměníky dělí na přímé a nepřímé. U nepřímých výměníku nedochází ke styku teplonosného média s procesním médiem nebo jiným teplonosným médiem. Nepřímé výměníky se dále dělí na trubkové, deskové a spirálové (Obrázek 2.1). Výměník použitý v této práci je nepřímý pájený deskový, a proto se následující text bude věnovat pouze tomuto typu výměníku a jeho porovnání s jinými.

Obrázek 2.1: Používané typy výměníků

Deskové výměníky jsou tvořeny z množství za sebou seřazených desek s těsným odstupem, které jsou k sobě staženy obvykle pomocí šroubů [2]. Tyto desky oddělují od sebe mezideskové prostory, což jsou místa, kterými protéká teplonosné médium. Desky jsou rovněž prolisovány tak, aby při proudění média docházelo k intenzivním turbulencím. Primární a sekundární médium protéká každé svým vlastním systémem kanálů a mezideskových prostor, přičemž není nikdy ve dvou sousedících mezideskových prostorech stejné médium (Obrázek 2.2).

Mezi výhody u tohoto typu patří zejména efektivita výměny tepla, dána protiproudým prouděním teplonosných médií, turbulencemi a prolisováním desek a u malých výměníku i kompaktní design. Při proudění teplonosných médií stejným směrem se efektivita výměny tepla prudce snižuje

(může být až poloviční). Malou nevýhodou deskových výměníku je nízká odolnost proti vysokým tlakům a nutnost čištění často zanášejících se desek.



Obrázek 2.2: Princip funkce deskového výměníku

Variantou deskových výměníku jsou pájené deskové výměníky (Obrázek 2.3). U tohoto typu konstrukce nahrazuje pájený spoj těsnění, přítlačné desky, stahovací šrouby a také rám. Pájené deskové výměníky se používají všude tam, kde je požadavek na pohodlí, malé rozměry, malý objem, dobrý poměr cena/výkon a poměrně vysokou účinnost. Tyto výměníky ovšem nejsou rozebíratelné a tudíž čištění jednotlivých desek je možné pouze proplachem chemikáliemi, která je nákladnější než mechanické rozebrání výměníku a očištění desek.

Obrázek 2.3: Pájený deskový výměník výrobce Alfa Laval

3 Komunikační prostředky

Komunikační prostředky jsou HW (Hardware – fyzicky realizovatelné části jako například PC apod.) a SW (Software – programy vytvořené v PC apod., pro ovládání HW) části, jejichž kombinace umožní komunikaci mezi dvěma a více body. HW část představuje převodník rozhraní AY1020, osobní počítač a sběrnice či datové kabely. SW části jsou například komunikační protokoly, což jsou standardizované způsoby kódování a přeposílání dat přes datové kabely. Další SW část představuje například aplikace v osobním či průmyslovém počítači, nebo program v PLC. Komunikace je v této práci prováděna za pomocí protokolu Modbus, rozhraní IO-Link, vývojového prostředí LabVIEW, PC a převodníku rozhraní AY1020.

3.1 LabVIEW

LabVIEW je softwarové vývojové prostředí od společnosti National Instruments a slouží jako programovací nástroj pro tvorbu různých aplikací, nejčastěji se ale využívá pro měření a testování reálných procesů. Jeho veliká výhoda oproti jiným konkurenčním prostředím je možnost individuálního řešení dané problematiky, tedy jednoduché přizpůsobení aplikace požadavkům zákazníka. V LabVIEW se programuje v tzv. G (grafickém) jazyku, na rozdíl od textových a strukturovaných jazyků (C apod.).

Obrázek 3.1: Logo vývojového prostředí LabVIEW

3.2 Modbus protokol

Modbus je komunikační protokol implementující 5 vrstev ISO/OSI modelu. Komunikace probíhá stylem klient – server, neboli Master – Slave [3]. Výhodou Modbusu je, že může být využitý na různých typech sběrnic a sítí, například na sériové lince RS-232 nebo síti Ethernet s podporou protokolu TCP/IP. Vzhledem k povaze bakalářské práce bude v následujícím textu probírána pouze verze Modbus/TCP.

Obrázek 3.2: Logo protokolu Modbus

3.2.1 Historie

Modbus byl vyvinutý společností Modicon, nyní součástí společnosti Schneider Electric, v roce 1979. V roce 2004 převedla firma Schneider Electric práva na Modbus v prospěch neziskové organizace Modbus-IDA. Ta vznikla spojením Modbus organization, neziskové obchodní skupiny sdružující uživatelé a dodavatelé zařízení na bázi Modbusu, a IDA group, která se angažuje ve vývoji standardů pro distribuované systémy využívající Ethernet TCP/IP jako univerzální komunikační standard.

Ještě téhož roku byl Modbus uznán organizací IEC jako veřejně dostupná specifikace. Jeho TCP modifikace byla připsána k normě IEC SC65C jako průmyslový Ethernetový protokol. Na jaře roku 2005 došlo k vytvoření specifikace pro komunikaci s CAN zařízeními. Poslední velkou událostí je připojení k vývoji bezdrátové specifikace založené na WirelessHART z května 2011 [4].

3.2.2 Popis protokolu

Modbus využívá pro předávání dat především tyto dva datové typy – celočíselný datový typ UINT (Unsigned Integer (16 bitů) – U16) a logickou hodnotu BOOL (Boolean (1 bit)). Celočíselný datový typ se používá v registrech pro předání analogových hodnot, v logickém datovém typu se předává dvoustavová informace (zapnuto - vypnuto, plné – prázdné apod.).

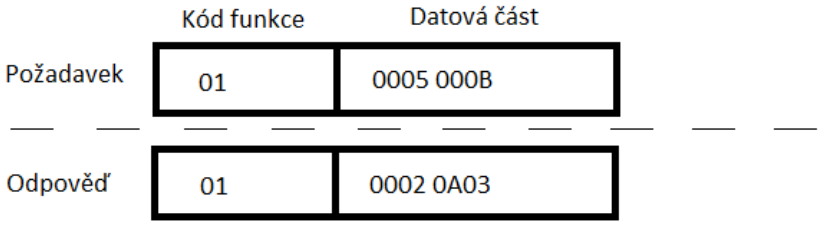
Základním prvkem Modbus komunikace je zpráva na úrovni PDU (Protokolová datová jednotka). Ta obsahuje:

- Kód funkce o velikosti 1 Byte
- Datovou část o proměnné velikosti

V definici Modbus protokolu je určeno 127 kódů funkcí [4], nicméně zařízení využívající Modbus protokol pro komunikaci nemusí všechny tyto kódy podporovat. Tyto podporované kódy se většinou nachází v dokumentaci k zařízení, tak ať uživatel ví, jaké zprávy může na cílové zařízení posílat. Nejčastěji používané funkce jsou:

- **Funkce Read Coils (0x01)**

Funkce Read Coils přečte hodnotu až dvou tisíc po sobě jdoucích binárních adres. V datové části požadavku je první binární adresa (2 Byte) a počet adres (2 Byte). V datové části odpovědi se pak nachází počet Bytů zprávy (N) a stavy jednotlivých adres (N Bytů). Stav 1 adresy je v prvním Bytu na pozici nejnižšího bitu. Pokud není počet adres v požadavku dělitelný osmi, je odpověď o jeden Byte delší. Prázdná místa v posledním Bytu jsou doplněna logickou nulou.



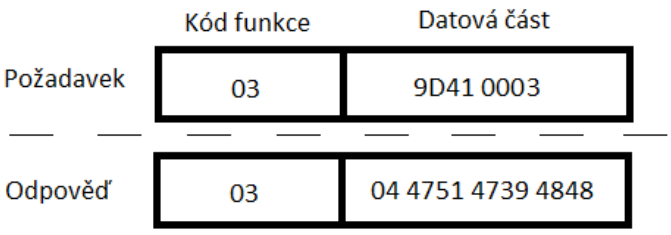
Obrázek 3.3: Příklad implementace funkce 01

Na obrázku (Obrázek 3.3) je vidět implementace funkce Read Coils v PDU. V datové části požadavku je počáteční adresa (0005 hex = 00005) a počet adres (000B hex= 11). V datové části odpovědi je počet bytů (0002 hex = 2 Byty) a stavy jednotlivých adres (0A = 00001010, 03= 00000011).

V tomto příkladu jdou vidět dvě věci. První je zápis počáteční adresy (00005). Maximální množství adres v Modbusu je 56536 a je pravidlem, že adresy (dále jen adresové prostory), na kterých je možné zavolat funkce Read Coils, Write Coil, nebo Write Multiple Coils se adresují od 00000 po 09999. Další adresové prostory budou popsány u příslušných funkcí. Druhou věcí je doplnění nulami po nejvyšší bit u posledního Bytu.

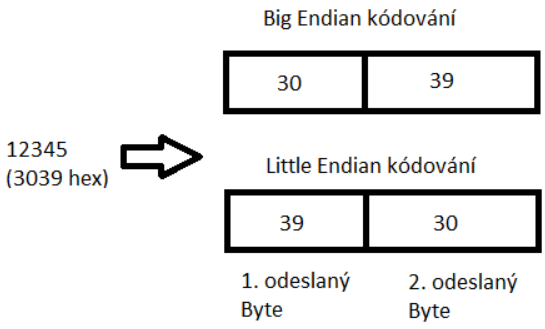
- **Funkce Read Holding Registers (0x03)**

  Funkce Read Holding Registers umožňuje čtení obsahu až 125 po sobě jdoucích registrů daného zařízení. V datové části požadavku je adresa prvního registru (2 Byty) a množství čtených registrů (2 Byty). V datové části odpovědi je délka zprávy (1 Byte) a obsah registrů (x Bytů). Obsah registrů je vyjádřen vždy dvěma Byty na registr.



Obrázek 3.4: Příklad implementace funkce 03

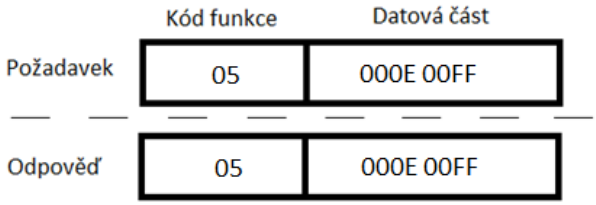
Na obrázku (Obrázek 3.4) je uveden příklad použití funkce Read Holding Registers. V datové části požadavku je adresa prvního čteného registru (9D41 hex = 40257) a počet registrů ke čtení (0003 hex = 3). V datové části odpovědi je délka zprávy (04 hex = 4 Byty) a hodnoty jednotlivých registrů (registr 40257 – 4751 hex = 18257, registr 40258 – 4739 hex = 18233, registr 40259 – 4848 hex = 18504). Při posílání dat z registrů se používá Big Endian kódování dat (Obrázek 3.5), takže se první posílá Byte s vyšší hodnotou a druhý Byte s nižší hodnotou. Pro funkce Read Holding Registers, Write Single Register se využívá adresový prostor ohraničený adresami 40000 a 49999.



Obrázek 3.5: Rozdíl mezi Big a Little Endian kódováním dat

- **Funkce Write Coil (0x05)**

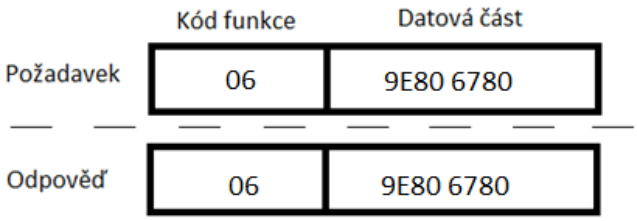
  Funkce Write Coil obsahuje v datové části požadavku adresu, na kterou má binární informaci zapsat (2 Byty) a danou hodnotu (2 Byty) – 00 00 pro logickou nulu a FF 00 pro logickou jedničku. Jakákoliv jiná hodnota neovlivní výstup. Odpověď je kopií požadavku. Příklad implementace funkce Write Coil je na obrázku níže (Obrázek 3.6).



Obrázek 3.6: Příklad implementace funkce 05

- **Funkce Write Holding register (0x06)**

  Funkce Write Holding register zapíše na adresu (2 Byty) hodnotu v šestnáctkové soustavě (2 Byty). Odpověď je přesná kopie požadavku. Příklad implementace je na obrázku níže (Obrázek 3.7).



Obrázek 3.7: Příklad implementace funkce 06

3.2.3 Implementace protokolu v LabVIEW

Ve vývojovém prostředí LabVIEW je pro zacházení s protokolem Modbus vytvořená knihovna funkcí. Vzhled funkcí se mění s verzí programu LabVIEW (Obrázek 3.8 vlevo LabVIEW 2015, vpravo LabVIEW 2014).

Práce s Modbus protokolem v LabVIEW se dá připodobnit k práci se souborem. Na začátku programu je nutné otevřít komunikační kanál vytvořením Master nebo Slave instance. Poté je možné na Modbus komunikaci volat konkrétní funkce. Jednotlivé funkce jsou představeny samostatnými bloky. LabVIEW umožňuje implementovat funkce Write Coil, Read Input Registers, Read Coils, Read Holding Registers, Read Discrete Inputs, Read/Write Multiple Registers, Write Single Register, Write Multiple Coils (liší se použitím v Master či Slave instanci), Write Multiple Registers (také se liší použitím

v Master či Slave instanci) a diagnostické funkce, které lze použít pouze v Master instanci, Read Device ID a Read Exception Status (Příloha 1).

Obrázek 3.8: Odlišnost vhledu funkcí v různých verzích LabVIEW

3.3 IO-Link

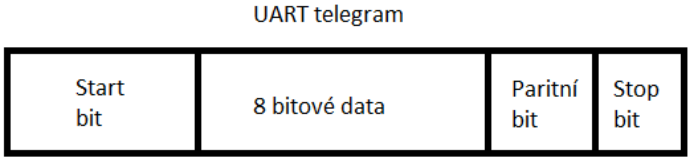
IO-link je komunikační rozhraní založené na komunikaci bod-bod nahrazující tradiční analogové proudové a analogové napěťové výstupy senzorů anebo také spínané výstupy senzorů [5]. V momentální době stojí za tímto nesíťovým komunikačním rozhraním celá řada výrobců z celého světa z důvodu jeho celkové propracovanosti. IO-Link je specifikován v normě IEC 61131-9.

Obrázek 3.9: Logo komunikačního rozhraní IO-Link

IO-Link představuje celou řadu zlepšení komunikačních možností se senzory a akčními členy. Z důvodu začlenění analogové komunikace do digitální lze připojené senzory nejen ovládat, ale také dálkově nastavovat. Tyto zařízení lze také připojit k jakékoliv sběrnici přes odpovídající rozhraní nazývané IO-Link Master, či Gateway (brána). V neposlední řadě patří mezi výhody jednoduché připojení, nastavení za běhu či rychlá výměna připojeného zařízení.

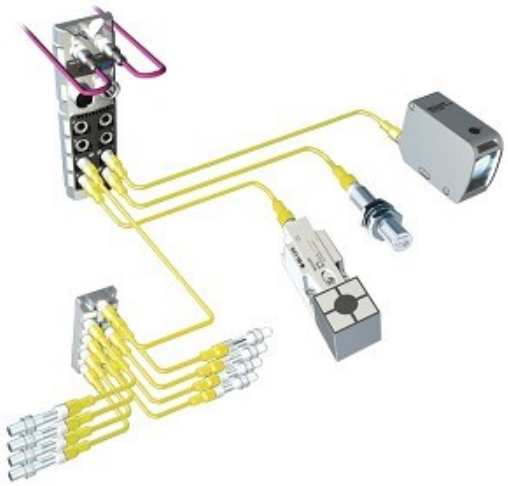
Jednou z nevýhod je, že jednotka zajišťující komunikaci s nadřazenou sběrnicí musí mít fyzikálně tolik vstupních portů, kolik je na ni připojených zařízení či senzorů (Obrázek 3.11). Další nevýhodou může a nemusí být maximální délka kabelu pro připojení senzoru, omezená na 20 metrů vyplývající z přenosu UART.

Komunikace mezi senzorem a nadřazenou IO-Link jednotkou je zajištěna osmi bitovým asynchronním přenosem dat UART a časovým multiplexingem. Osmibitová digitalizovaná data bez multiplexingu se nazývají telegramy (Obrázek 3.10).



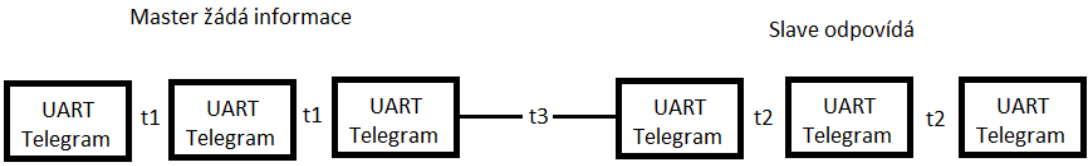
Obrázek 3.10: Složení UART telegramu

Množství telegramů s časovým rozestupem (časovým multiplexingem) mezi Master a Slave zařízením se nazývají rámce, anglicky frame (Obrázek 3.12). Časový multiplexing (čas mezi koncem jednoho telegramu a začátkem nového telegramu) má rozdílnou hodnotu pro Master (t1) a Slave (t2) zařízení. V jednom rámci rozlišujeme ještě pauzu mezi koncem odesílání Master telegramů a začátkem odesílání Slave telegramů (t3). Tato pauza musí být větší, než čas t1 a odlišný od času t2. IO-Link rámce mají standardizovanou strukturu, podle které určujeme data přenášená konkrétním rámcem:



Obrázek 3.11: IO-Link Master zařízení s připojenými senzory a rozbočovačem

- Procesní data – data naměřená senzorem
- Data na vyžádání – nastavení, parametry senzoru
- Přímé parametry/diagnostická data – takto se nastavují senzory
- Události – chyby či neobvyklé události



Obrázek 3.12: Časové rozložení rámce

Tyto rámce se následně předávají nadřazené SW aplikaci uvnitř IO-Link zařízení. Tato aplikace většinou připravuje data a zařizuje komunikaci po nadřazené sběrnici (Průmyslový Ethernet, Profibus, Fieldbus apod,)

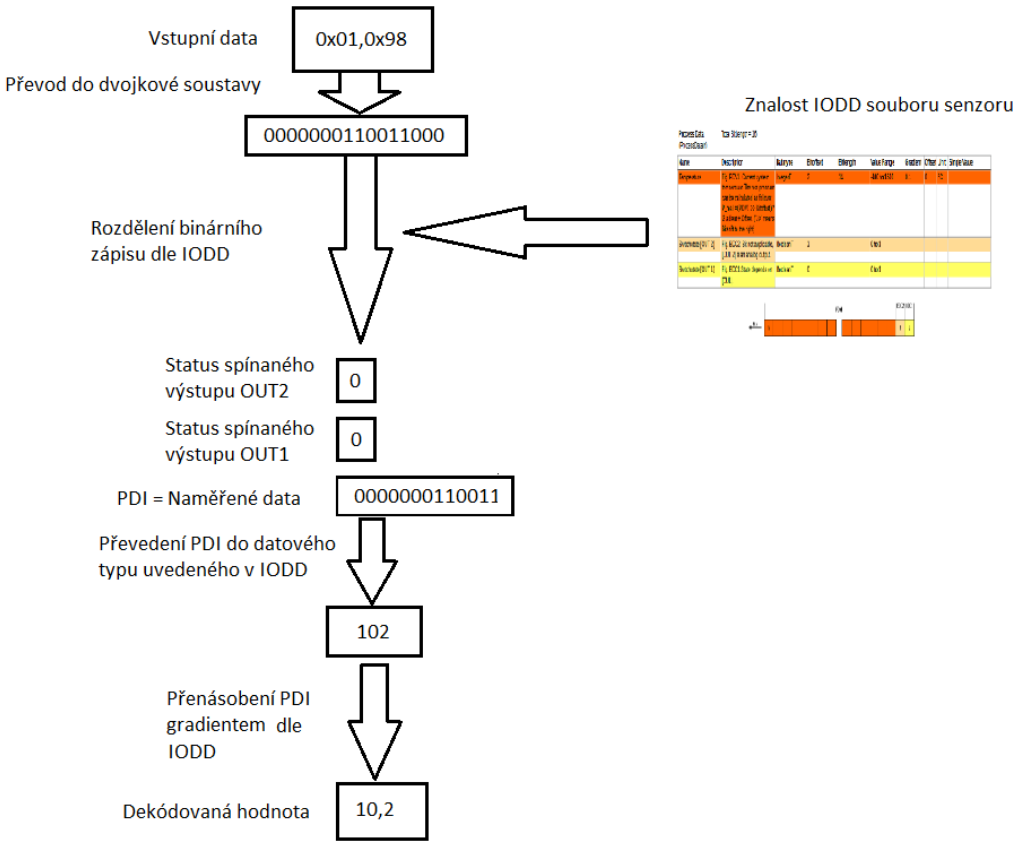
Senzor připojený k IO-Link Master jednotce se identifikuje pomocí tzv. IODD (IO Device Description) souboru, který v sobě nese informace o zakódování výstupních parametrů, typu senzoru, velikosti výstupní zprávy a dalších nastavitelných i nenastavitelných parametrech.

4 Hardware

Laboratorní model pro demonstraci výměny tepla se na pracovišti nacházel již dříve, takže vlastní výběr použitých přístrojů a senzorů byl znemožněn. V následující kapitole jsou probrány senzory a jejich případné nastavení. Nastavování jednotlivých senzorů je téměř totožné a lze je nastavovat přes programovatelná tlačítka a LED displej, nebo také přes posílání ISDU (Indexed Service Data Unit) příkazů z počítače, či PLC. Hladinoměr LMT 121 postrádá displej a proto jej lze nastavit pouze pomocí ISDU příkazů.

ISDU příkazy jsou strukturované požadavky používané technologií IO-Link pro nastavování parametrů u jednotlivých senzorů. Lze takto nastavit třeba měřící rozsah, citlivost a jiné funkce dostupné u senzoru. Každý senzor definuje své ISDU zprávy.

Senzory odesílají všechny naměřené hodnoty na IO-Link zakódované v jednom rámci, a proto je nutné pro získání pouze jedné hodnoty tento rámec rozkódovat. K tomu slouží bitový posun a gradient. Daná zpráva se tedy nejdříve převede na binární pole s nejvyšším bitem na pozici 0, provede se bitový posun vpravo a zleva se doplní nulami. Bitový posun slouží k získání pouze jedné informace z rámce. Pro získání všech informací z rámce lze vstupní zprávu rozdělit na různě dlouhé úseky. Výsledný binární řetězec (řetězce) se převede na decimální hodnotu a nakonec se vynásobí gradientem (Obrázek 4.1). Gradient a bitový posun jsou u každého údaje jiné a budou popsány u jednotlivých senzorů [6].



Obrázek 4.1: Princip dekódování dat ze senzoru

4.1 Senzor teploty IFM TN2531

Elektronický teplotní senzor TN2531 (Obrázek 4.2) vyrábí německá společnost IFM. Používá se pro měření teploty kapalných a plynných medií do tlaků 300 barů. Měřícím elementem je teploměr Pt1000 ve třídě B s měřícím rozsahem -40 až 150 °C. Senzor se připojuje k procesu pomocí adaptéru s metrickým závitem M18 x 1,5 a minimální průměr trubky s měřeným médiem musí být větší, než je instalační délka senzoru (45 mm). Minimální hloubka ponoru pro zajištění korektního měření je 12 mm.

Výstup senzoru obstarává čtyř pinový konektor a komunikační rozhraní IO-Link 1.1. Senzor kóduje výstupní data pro IO-Link. Tabulka kódování je uvedena pod odstavcem (Tabulka 4.1).

Obrázek 4.2: Senzor TN2531 na pracovišti

Senzor má množství nastavitelných parametrů. Důležitým nastavitelným parametrem je funkce LOCK, která znemožňuje nastavení přístroje tlačítky. Další nastavitelné parametry může být chování spínaných výstupů při různých událostech.

Tabulka 4.1: Kódování výstupních dat senzoru typu TN2531 dle Obrázek 4.1

Údaj	Bitový posun	Gradient	Počet bitů
Teplota	2	0,1	14
Status výstupu OUT2	1	-	1
Status výstupu OUT1	0	-	1

4.2 Senzor hladiny IFM LMT121

Elektronický senzor hladiny LMT121 (Obrázek 4.3) je dalším použitým produktem od společnosti IFM. Měří hladinu v kapalných, pastových a práškových mediích a lze je využít pro kontinuální měření, limitní hlídání maximální i minimální hladiny a také je vybaven dvěma

nastavitelnými spínacími body, které mohou posloužit pro detekci dvou odlišných medií (například detekce vody s pěnou).

Obrázek 4.3: Senzor LMT121 na pracovišti

Princip měření spočívá ve spektrálním měření impedance od 50 do 200 MHz. Pro různá média existují různé charakteristiky.

LMT121 rovněž využívá čtyř pinový výstupní konektor a IO-Link 1.1. Výstupní data pro IO-Link kóduje, kódování je uvedeno v tabulce pod odstavcem (Tabulka 4.2). Procesní připojení obstarává patice s dvoupalcovým trubkovým závitem a délka sondy je 11 mm.

LMT 121 nemá digitální displej, nýbrž dvě LED diody indikující přítomnost kapaliny a z tohoto důvodu jej lze nastavovat pouze pomocí ISDU příkazů z připojeného PC či PLC.

Tabulka 4.2: Kódování výstupních dat senzoru typu LMT121 dle Obrázek 4.1

Údaj	Bitový posun	Gradient	Počet bitů
Hladina	2	1	14
Status výstupu OUT2	1	-	1
Status výstupu OUT1	0	-	1

4.3 Senzor průtoku IFM SM9000

Magneticko-induktivní senzor proudění SM9000 (Obrázek 4.4) od IFM je posledním typem senzoru použitým v laboratorním modelu pro měření předání tepla. Senzor je vybaven displejem, kde se zobrazují aktuální procesní hodnoty, výstupem senzoru jsou dva nastavitelné signály (OUT1, OUT2).

Tento přístroj je schopný měřit aktuální průtočné množství, celkový průtok zařízením od posledního restartu a také aktuální teplotu média a rovněž je schopný detekovat prázdnou trubici. Průtok se měří magneticko-indukčním principem. Ten spočívá v měření vychýlení iontů magnetického pole.

Všechny analogové výstupní signály jsou z výroby nastaveny jako proudové (4-20mA). Přístroj je připojen dvoupalcovým trubkovým závitem a má také čtyř-pinový konektor a IO-Link 1.1. Senzor rovněž kóduje výstupní data pro IO-Link, tabulka kódování je uvedena pod odstavcem (Tabulka 4.3).

Obrázek 4.4: Senzor SM9000 na pracovišti.

Senzor SM9000 je jediným použitým senzorem, který má možnost simulovat měřené hodnoty. Simulace jde využít například při ověření funkčnosti senzoru. Po ukončení simulace se nenávratně ztratí všechny simulované hodnoty.

Tabulka 4.3: Kódování výstupních dat senzoru typu SM9000 dle Obrázek 4.1

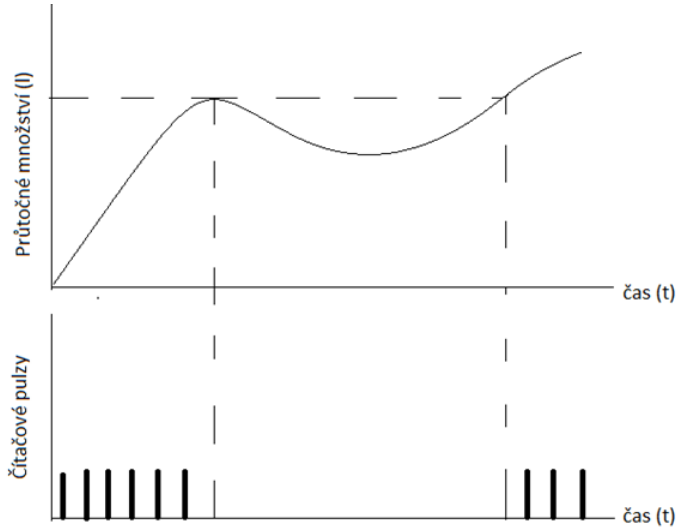
Údaj	Bitový posun	Gradient	Počet bitů
Celkový průtok	32	1	32
Momentální průtok	16	0,1	16
Teplota	2	0,1	14
Status výstupu OUT2	1	-	1
Status výstupu OUT1	0	-	1

4.3.1 Měření aktuálního průtočného množství

Informaci o aktuálním průtočném množství poskytuje senzor jako frekvenční signál. Senzor rovněž detekuje, zda směr proudění kapaliny odpovídá směru proudění vyznačenému na přístroji. V případě, kdy tomu tak není, se procesní hodnoty zobrazují na displeji se záporným znaménkem a tyto data následně nejsou zpracovány pro výstupní signály.

4.3.2 Měření celkového průtočného množství

Senzor kromě snímání aktuálního průtočného množství obsahuje i interní čítač celkového množství. Ten generuje čítací impulzy pouze při navýšení momentálního celkového průtočného množství. V případě, že by bylo proudění média v senzoru proti vyznačenému směru, čítač odečítá momentální hodnoty od celkového množství. Po opětovném otočení směru proudění se generují pulzy až od dosažení celkového průtoku před odečítáním (Obrázek 4.5). Celkové průtočné množství je čítáno až do resetu čítače, který může být proveden manuálně, nebo může být časově řízený. Obě tyto možnosti jsou nastavitelné přímo v menu senzoru, ale také přes ISDU příkazy.



Obrázek 4.5: Princip generování pulzu pro čítání průtočného množství

U tohoto senzoru lze také nastavit maximální hodnotu průtočného množství, tato informace může být posléze hlídána dvěma způsoby. První způsob představuje generování výstupního impulzu při dosažení nastaveného množství. Další impulzy nejsou generovány, dokud není čítač resetován a není opětovně dosažena nastavená hodnota.

Druhý způsob představuje dvě možnosti hlídání množství. Jednou možností je časové hlídání množství. Pokud množství přesáhne za stanovený čas nastavenou hodnotu, spínaný výstup (OUT1) sepne a zůstane sepnut, dokud nebude čítač resetován. Pokud nedojde k přesažení maximálního množství, bude čítač po nastavené době resetován automaticky. Druhou možností je obyčejné hlídání množství. Pokud toto množství přesáhne stanovenou mez, sepne výstup (OUT1) a také zůstává sepnut, dokud není čítač resetován.

4.3.3 Měření teploty

Výstupem senzoru při měření teploty mohou být dva signály a to spínací signál pro mezní hodnotu teploty nebo teplotně úměrný analogový signál. Při využití IO-Link rozhraní se tato hodnota posílá spolu s ostatníma měřenýma hodnotama.

4.3.4 Rozpoznání prázdné trubice

Toto nastavení je volitelné, dá se nastavit pro časově závislé rozpoznání prázdné trubice, časově nezávislé rozpoznání prázdné trubice, nebo může být toto nastavení úplně vypnuto. Pokud je zapnuto, a trubice je prázdná, rozsvítí se na displeji SEnS a proudění se nastaví na nulu. Senzor může tuto informaci odesílat v jednom ze spínaných výstupů.

4.4 Převodník rozhraní IFM AY1020

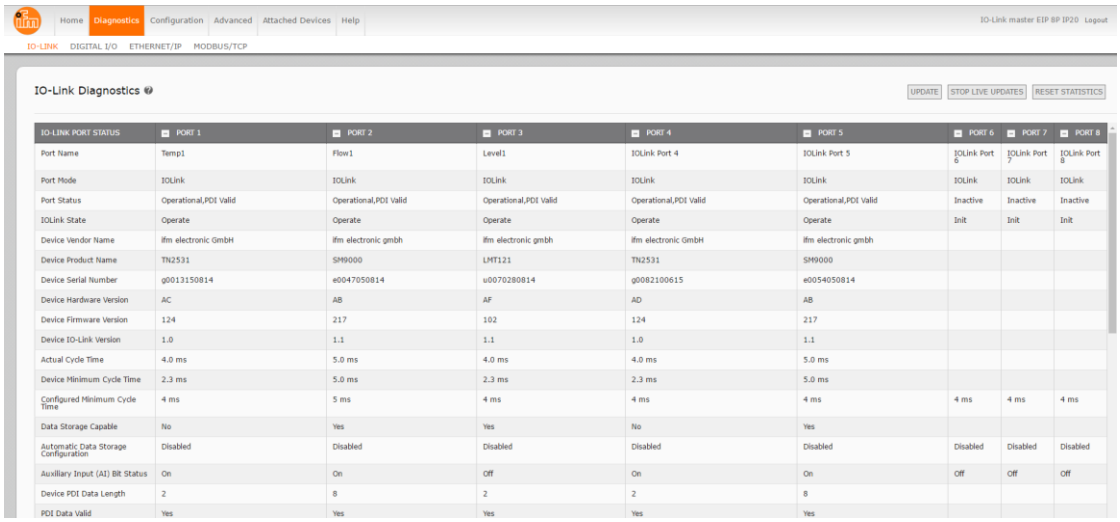
Komunikační jednotka AY1020 (Obrázek 4.6) představuje vstupně výstupní převodník mezi rozhraním IO-Link na straně senzorů a Ethernet na straně PLC (PC). Na straně rozhraní IO-Link představuje Master jednotku, na straně Modbus rozhraní poté Slave jednotku.

Převodník rozhraní AY1020 se skládá z 8 IO-Link portů, 2 IO-Link portů určených pro realizaci digitálních vstupů/výstupů a 2 napájecích portů, z nichž může být zapojen právě jeden, a dvou zásuvek RJ45 pro rozhraní Ethernet. Převodník rozhraní AY1020 je schopný komunikovat prostřednictvím dvou

protokolů, EtherNet/IP a Modbus/TCP. V této práci se používá protokol Modbus/TPC, proto se dále nebudu zabývat charakteristikou komunikace přes protokol EtherNet/IP.

Obrázek 4.6: Převodník rozhraní AY1020 na pracovišti

Převodník rozhraní AY1020 lze okamžitě po připojení k osobnímu počítači naprogramovat přes web server (Obrázek 4.7) přístupný z internetového prohlížeče připojeného počítače po zadání adresy 196.168.1.250. Tato adresa je nastavená výrobcem a lze jí libovolně změnit. Do zmíněného web serveru je nutné nahrát IODD soubory všech připojených senzorů a nastavit IO-Link porty dle připojených senzorů (velikost odesílané zprávy, frekvence obnovy dat, atd.). Ve web serveru jsou vidět i poslední přijaté či odeslané data převodníkem rozhraní.

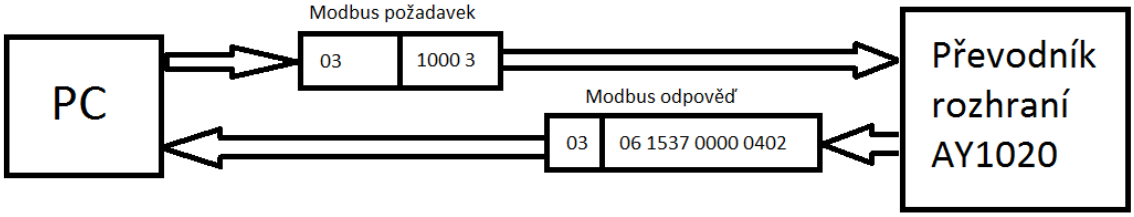


IO-LINK PORT STATUS	PORT 1	PORT 2	PORT 3	PORT 4	PORT 5	PORT 6	PORT 7	PORT 8
Port Name	Temp1	Flow1	Level1	IOLink Port 4	IOLink Port 5	IOLink Port 6	IOLink Port 7	IOLink Port 8
Port Mode	IOLink	IOLink	IOLink	IOLink	IOLink	IOLink	IOLink	IOLink
Port Status	Operational,PDI Valid	Operational,PDI Valid	Operational,PDI Valid	Operational,PDI Valid	Operational,PDI Valid	Inactive	Inactive	Inactive
IOLink State	Operate	Operate	Operate	Operate	Operate	Init	Init	Init
Device Vendor Name	ifm electronic GmbH	ifm electronic gmbh	ifm electronic gmbh	ifm electronic GmbH	ifm electronic gmbh			
Device Product Name	TN2531	SM9000	LMT121	TN2531	SM9000			
Device Serial Number	g0013150814	e0047050814	u0070280814	g0082100615	e0054050814			
Device Hardware Version	AC	AB	AF	AD	AB			
Device Firmware Version	124	217	102	124	217			
Device IO-Link Version	1.0	1.1	1.1	1.0	1.1			
Actual Cycle Time	4.0 ms	5.0 ms	4.0 ms	4.0 ms	5.0 ms			
Device Minimum Cycle Time	2.3 ms	5.0 ms	2.3 ms	2.3 ms	5.0 ms			
Configured Minimum Cycle Time	4 ms	5 ms	4 ms	4 ms	4 ms	4 ms	4 ms	4 ms
Data Storage Capable	No	Yes	Yes	No	Yes			
Automatic Data Storage Configuration	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
Auxiliary Input (AI) Bit Status	On	On	Off	On	On	Off	Off	Off
Device PDI Data Length	2	8	2	2	8			
PDI Data Valid	Yes	Yes	Yes	Yes	Yes			

Obrázek 4.7: Webserver převodníku rozhraní AY1020

Při znalosti Modbus protokolu (Kapitola 3.2.2) je nutné znát pro komunikaci kód požadované funkce i adresu počátečního registru a požadované množství registrů. S těmito parametry lze s převodníkem rozhraní AY1020 plnohondnotně komunikovat, nastavovat nebo číst data. Komunikace

PC s převodníkem rozhraní AY1020 přes Modbus protokol je popsána na následujícím příkladu (Obrázek 4.8). Pro jednoduchost jsou všechny předávané parametry popsány v desítkové soustavě.



Obrázek 4.8: Příklad Modbus komunikace mezi PC a převodníkem rozhraní AY1020

Modbus požadavek obsahuje kód funkce, počáteční adresu registru a počet adres (registrů). Převodník rozhraní AY1020 podporuje pouze omezené množství Modbus kódů funkcí a to:

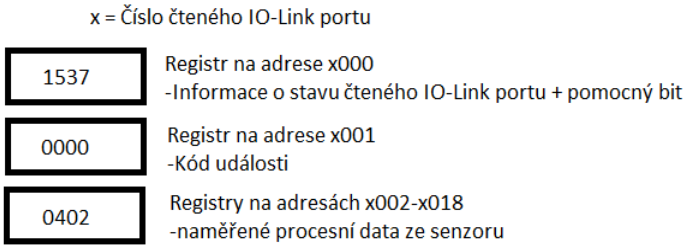
- Read Holding Registers (Čtení více Registrů) – kód funkce 0x03
- Write Single Register (Zápis dat do registru) – kód funkce 0x06
- Write Multiple Registers (Zápis dat do více registrů) – kód funkce 0x10
- Read/Write Holding Registers (Čtení/Zápis dat do více registrů) – kód funkce 0x17

Adresy jsou v převodníku rozhraní AY1020 rozděleny podle portů (Tabulka 4.4Tabulka 4.4), počet čtených adres na portu se odvíjí od připojeného senzoru. Nejmenší počet čtených adres (registrů) je rovný třem. Důvod čtení minimálně tří registrů bude vysvětlen u popisu Modbus odpovědi v tomto příkladu.

Tabulka 4.4: Rozdělení adresového prostoru v převodníku rozhraní AY1020

Číslo IO-Link portu	Přidělené adresy
1	1000-1999
2	2000-2999
3	3000-3999
4	4000-4999
5	5000-5999
6	6000-6999
7	7000-7999
8	8000-8999

Modbus odpověd obsahuje kód funkce, množství následujících Bytů v odpovědi a výstupní procesní data z převodníku (neboli PDI – proces data input). PDI se skládá z informace o stavu čteného IO-Link portu (1 Byte), informace o připojeném pomocném bitu IO-Link portu (1 Byte), kódu případné události (2 Byty) a vlastní procesní data (až 32 Bytů). Tyto data jsou sestaveny do registrů o velikosti 2 Byty (Obrázek 4.9). Pokud je odesílaná procesní hodnota hodnota větší než 65535, je tato hodnota binárně rozdělena na dvě části po 16 bitech (2 Byty) a odeslána ve dvou registrech o velikost 2 Byty.



Obrázek 4.9: Sestavení PDI

Stav IO-Link portu (první čtený registr) může nabývat 5 možných hodnot podle stavu příslušných bitů.

0. Inicializační bit - Pokud je na tomto bitu logická nula, neprobíhá na senzoru inicializace komunikace, pokud je zde logická jednička, inicializační proces je aktivní
1. Provozní bit - Pokud je na tomto bitu logická nula, není připojené zařízení provozuschopné, pokud je zde logická jednička je připojené zařízení provozuschopné. Další stavy jsou určeny stavem chybového bitu.
2. Platnostní bit – Pokud je zde logická nula, data z připojeného senzoru nejsou platná. Pokud je zde logická 1, posílá připojený senzor platná data.
3. Chybový bit – Pokud je zde logická nula, není detekována žádná chyba. Pokud je zde logická 1, detekuje se chyba. Ta se rozděluje dále v závislosti na stavu provozního bitu.
    a. Pokud je na provozním bitu logická jednička, byla na portu detekována malá chyba, způsobena dočasnou ztrátou komunikace s připojeným senzorem nebo opravitelnou HW či SW chybou.
    b. Pokud je na provozním bitu logická nula, byla na konkrétním IO-Link portu detekována velká chyba, způsobena nevratnou ztrátou komunikace s připojeným senzorem nebo neopravitelnou HW či SW chybou.
4. Další bity jsou rezervovány a jsou nulové.

V druhém Bytu prvního čteného registru je informace o nastavení digitálních vstupů na čteném IO-Link portu (pomocný bit). Zapnutá možnost digitálních vstupů umožňuje vedle klasické funkce IO-Linku i číst stavy jednotlivých připojených svorek (Obrázek 4.10). Funkčnost je rozlišena stavy jednotlivých bitů:

0. Status bit – Označuje zda je pomocný bit zapnutý (1) nebo vypnutý (0). Pokud je vypnutý, všechny další bity jsou nulové.
1. Bity 1-3 jsou rezervovány a jsou nulové.
4. L+ status bit – Stav L+ svorky čteného IO-Link portu.
5. DI status bit – Stav DI svorky čteného IO-Link portu.
6. L- status bit – Stav L- svorky čteného IO-Link portu.
7. C/Q status bit – Stav vstupní/výstupní svorky čteného IO-Link portu.

Obrázek 4.10: Zobrazení svorek u převodníku rozhraní AY1020

Následuje kód události, který je při běžném provozu nulový. V dalších registrech jsou procesní data. Procesní data mají různou velikost podle připojeného přístroje. Velikost těchto dat nabývá vždy sudého počtu tak, ať mohou být tyto data odeslány v registrech o velikosti 2 Byty. Tyto data jsou zakódovaná podle připojeného přístroje (Tabulka 4.1 až Tabulka 4.3).

5 Návrh řešení automatizace měření veličin laboratorního modelu

Po nastudování vybrané literatury a domluvě s vedoucím práce byly stanoveny následující cíle:

- Pro vývoj aplikace bude použito programovací prostředí LabVIEW od společnosti National Instruments.
- Měřící aplikace bude obsahovat zobrazení aktuálních hodnot, grafy s historií, logování dat a výpočet tepelného výkonu
- Je nutné zamezit manipulaci nastavení senzorů a aplikace studenty

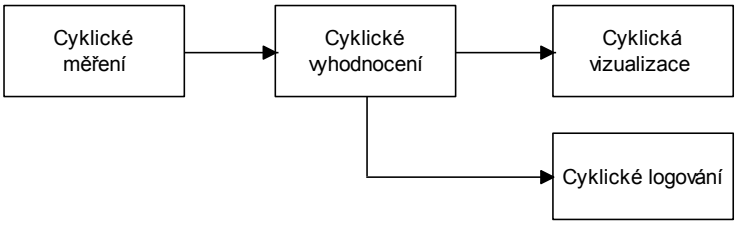
Vývojové prostředí LabVIEW se intenzivně využívá v průmyslové praxi pro vývoj aplikací pro automatizaci měření a testování. Dokáže rovněž využít zásuvku RJ45 v PC, a proto pro samotnou komunikaci s procesem (respektive s převodníkem rozhraní AY1020) není zapotřebí jiného zařízení (Hardware).

V měřící aplikaci se budou zobrazovat všechny aktuálně naměřené hodnot procesních veličin (Teploty, průtoky apod.) spolu s přesnou hodnotou hustoty, která je teplotně závislá a vypočtenou hodnotou tepelného výkonu obou okruhů. Dále se budou zobrazovat grafy teplot chlazeného a ohřívaného okruhu a graf minutových průtoků s pevně stanovenou historií.

Zamezení manipulace s procesním pracovištěm bude zajištěno uzamčením senzorů a umístěním ovládání prvků do rozvaděčů. Zamezení manipulace s nastavením aplikace bude zajišťovat chránění přístupu k nastavení.

Na základě všech uvedených parametrů byly vytvořeny 3 konceptuální řešení, lišící se především způsobem předávání dat mezi měřicí a vyhodnocovací částí, logovací částí a vizualizační částí. Tato řešení se následně realizovala za účelem porovnání a výběru nejlepšího řešení.

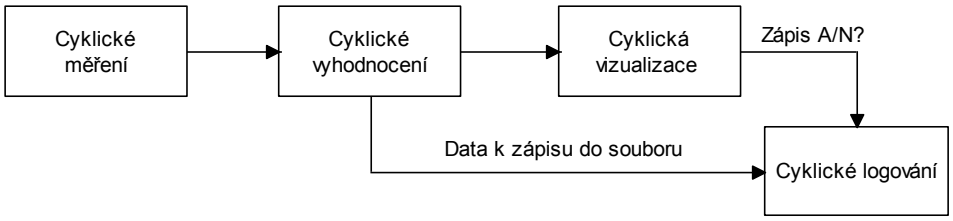
Kromě předávání dat fungují všechny koncepty principově stejně. Všechny obsahují ty nejdůležitější logické části – čtení z převodníku rozhraní AY1020, vyhodnocení parametrů, vizualizaci a zápis hodnot (Obrázek 5.1).



Obrázek 5.1: Blokové schéma principu funkce aplikace

5.1 První koncept

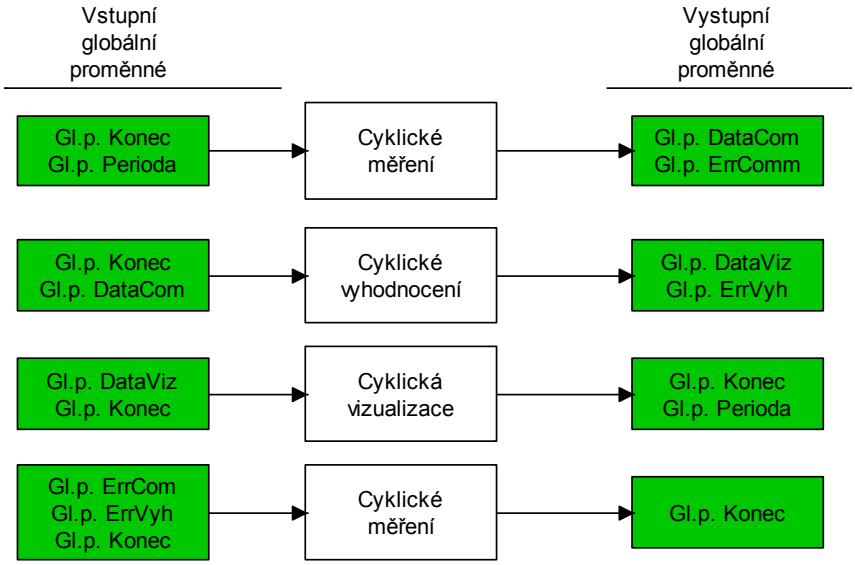
První koncept je vytvořen jako jednoduchá a rychlá varianta pro měření, vyhodnocování a předávání dat. Funguje prostřednictvím jednoho programu, který obsahuje všechny nutné funkce (Obrázek 5.2). Program běží ve smyčce WHILE s podmínkou ukončení stiskem tlačítka na uživatelském rozhraní. Program nepracuje paralelně, lze jej tedy označit jako sekvenční.



Obrázek 5.2: Blokové schéma prvního konceptu.

5.2 Druhý koncept

Druhý koncept představuje složitější provedení architektury. Zvýšená složitost spočívá v předávání dat mezi jednotlivými logickými částmi přes globální proměnné. Znamená to, že každý logický blok (měření, vyhodnocení, zápis, vizualizace) běží ve vlastním podprogramu. Celkem se tedy tento koncept sestává z jednoho programu obsahujícího čtyři samostatně běžící podprogramy ve smyčkách WHILE se dvěma podmínkama ukončení běhu (Obrázek 5.3). První podmínkou je vzniklá chyba vyhodnocená jako kritická a druhou podmínkou je stisk tlačítka na čelním panelu.



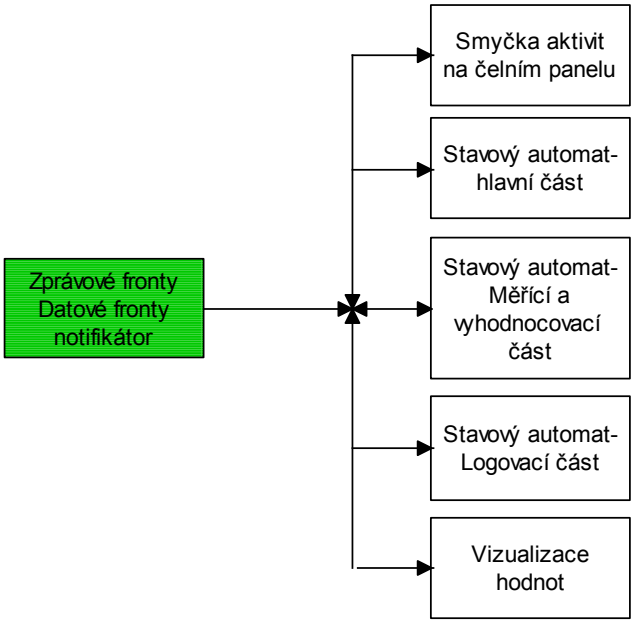
Obrázek 5.3: Blokové schéma druhého konceptu

5.3 Třetí koncept

Třetí koncept je realizačně nejsložitější. Funguje sice v rámci jednoho programu, který ale obsahuje pět smyček WHILE. Vychází ze tzv. Message Structure (Struktury založené na zprávách) a data mezi jednotlivými smyčkami se předávají ve frontách a notifikátorech (Obrázek 5.4). Fronty v LabVIEW jsou FIFO fronty, do kterých lze zapisovat a vyčítat z rozdílných míst programu, bez nutnosti ukončení některé ze smyček či nechtěného přepsání této fronty v jiné smyčce.

FIFO fronta se vyznačuje tím, že první zapsaný prvek do této fronty je také prvním prvkem vyňatým z této fronty. LabVIEW Notifikátor slouží rovněž k předání dat mezi paralelně běžícími

smyčkami, nicméně u notifikátoru vzniká problém při zápisu dvou prvků. Při následném výběru je v notifikátoru pouze jeden prvek a to ten naposled zapsaný.



Obrázek 5.4: Blokové schéma třetího konceptu

6 Realizace práce

Realizace práce obnášela tvorbu všech vizualizačních, měřících a komunikačních programů, vytvořených v rámci této práce a také všechny mechanické práce potřebné pro zprovoznění procesu.

Tvorba programů probíhala současně spolu s rozšiřováním znalostí o procesu, programovacím prostředí a komunikaci senzorů s PC od vytvoření simulátoru přes zprovoznění komunikace s PC po vlastní tvorbu uživatelského rozhraní. Tvorba byla dále ovlivněna pokyny vedoucího této práce a rovněž pokyny správce laboratoře, v niž se měřící pracoviště nachází.

Funkční kódy a podprogramy použité při realizaci všech konceptů budou popsány u realizace finálního produktu. Kódy a podprogramy použité pouze u některých konceptů budou u těchto konceptů také popsány.

Komunikace senzorů s převodníkem rozhraní AY1020 probíhá přes protokol IO-Link, z převodníku rozhraní se pro komunikaci s PC používá protokol Modbus/TCP.

6.1 Mechanické připojení pracoviště a nastavení komunikace

K realizace měřícího procesu bylo třeba elektricky připojit jednotlivé senzory k převodníku rozhraní AY1020 a převodník Ethernetovým kabelem k PC. Komunikace mezi převodníkem rozhraní AY1020 a PC je zajištěna protokolem Modbus. Zapojení převodníku rozhraní je zobrazeno na následujícím obrázku.

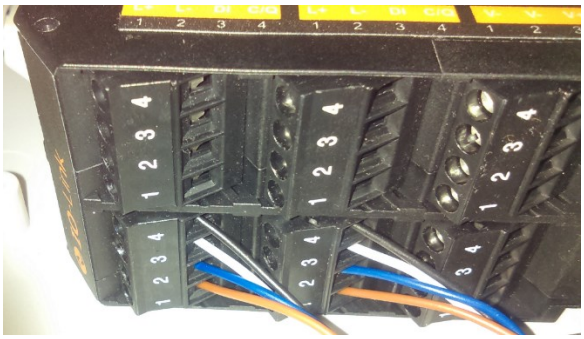
Následně bylo nastaveno chování převodníku rozhraní AY1020 přes web server. Nejdříve se musely nahrát IODD soubory, který automaticky vyhledaly odpovídající připojené senzory (Obrázek 6.1). Následně bylo třeba nastavit velikost PDI z AY1020 u portů dle použitých senzorů, nastavit všechny využívané porty do Slave modu a číslo všech používaných portů nastavit jednotně tak, ať není třeba při každém čtení měnit parametry připojení.

Obrázek 6.1: Nahrávání IODD souborů do web serveru AY1020

Výsledné zapojení (Obrázek 6.2) a nastavení jednotky AY1020 tedy vypadá následovně:

- Port 1: Senzor TN2531 okruhu s chlazenou vodou
  - Velikost odesílané PDI – 8B (4 Holding registry) – data jsou ale pouze ve 3 registrech
  - Byty PDI nejsou přehozeny
  - Data jsou odesílány na jednotku AY1020 každé 4 ms
- Port 2: Senzor SM9000 okruhu s chlazenou vodou
  - Velikost odesílané PDI – 16B (8 Holding registrů) – data jsou uložena pouze v 6 registrech
  - Byty PDI nejsou přehozeny

    - Data jsou odesílány na jednotku AY1020 každých 5 ms
- Port 3: Senzor LMT121 nádrže s chlazenou vodou
    - Velikost odesílané PDI – 8B (4 Holding registry) – data jsou uložena pouze v 3 registrech
    - Byty PDI nejsou přehozeny
    - Data jsou odesílány na jednotku AY1020 každé 4 ms
- Port 4: Senzor TN2531 okruhu s ohřátou vodou
    - Velikost odesílané PDI – 8B (4 Holding registry) – data jsou uložena pouze v 3 registrech
    - Byty PDI nejsou přehozeny
    - Data jsou odesílány na jednotku AY1020 každé 4 ms
- Port 5: Senzor SM9000 okruhu s ohřátou vodou
    - Velikost odesílané PDI – 16B (8 Holding registrů) – data jsou uložena pouze v 6 registrech
    - Byty PDI nejsou přehozeny
    - Data jsou odesílány na jednotku AY1020 každých 5 ms



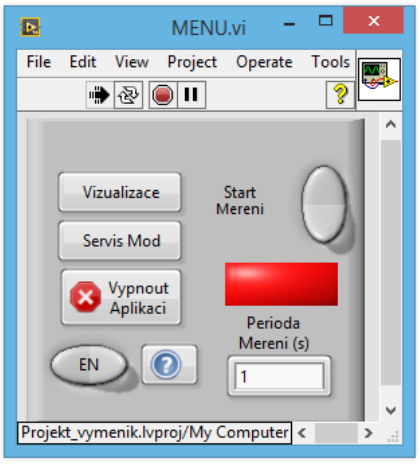
Obrázek 6.2: Zapojení senzorů do převodníku rozhraní AY1020

Pro zabránění nechtěných zásahů do nastavení senzorů byly všechny senzory s displejem uzamčeny. Pro nastavení senzorů pomocí displeje je nutné konkrétní senzor odemknout, jinak lze senzory nastavit přes rozhraní IO-Link z webserveru připojeného počítače.

6.2 Realizace prvního konceptu

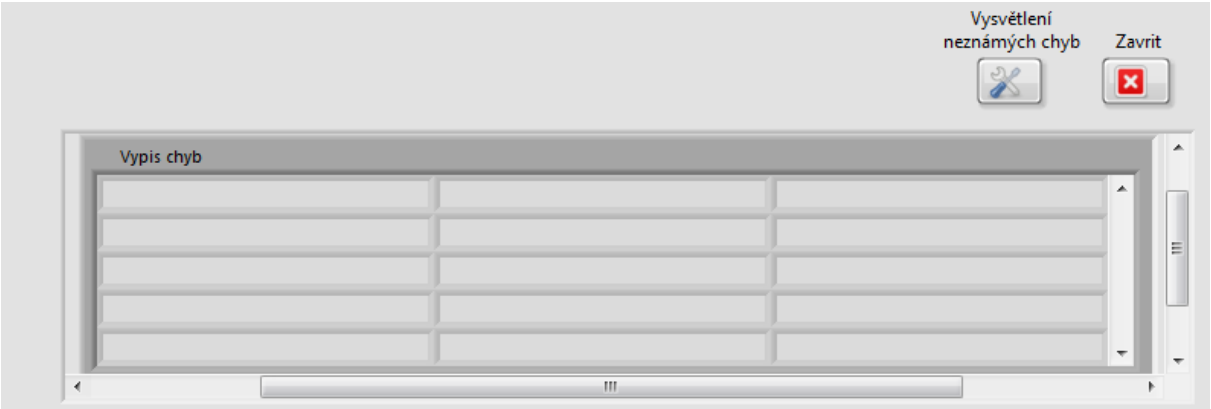
Realizací prvního konceptu vznikla aplikace Menu.vi (Obrázek 6.3). Po spuštění Menu.vi, šel vybrat čelní panel některého z podprogramů, který se měl zobrazit – Servisní podprogram s výčtem vzniklých chyb v aplikaci i na pracovišti, Vizualizační podprogram, kde se zobrazovaly aktuální naměřené hodnoty. Z tohoto programu se spouštělo měření, nastavovala hodnota periody měření, měnil jazyk aplikace a ukončoval se běh aplikace.

Tento koncept lze rozdělit do dvou propojených částí - cyklické zpracování naměřených dat a chyb a samostatné části pro zpracování reakcí na aktivity čelního panelu. Reakce na aktivitu čelního panelu řeší pouze spuštění jiného čelního panelu, nastavení jazykových modifikací.



Obrázek 6.3: Čelní panel VI Menu.vi prvního konceptu

Cyklické zpracování chyb umožnuje zobrazení všech chyb v podprogramu KnownErrors.vi (Obrázek 6.4), které se vyskytly v procesu měření v průběhu jedné iterace smyčky. Chyby jsou v tomto případě rozděleny na popsané a neznámé. Tím lze vyskytlou chybu identifikovat a eliminovat příčinu chyby za běhu aplikace.

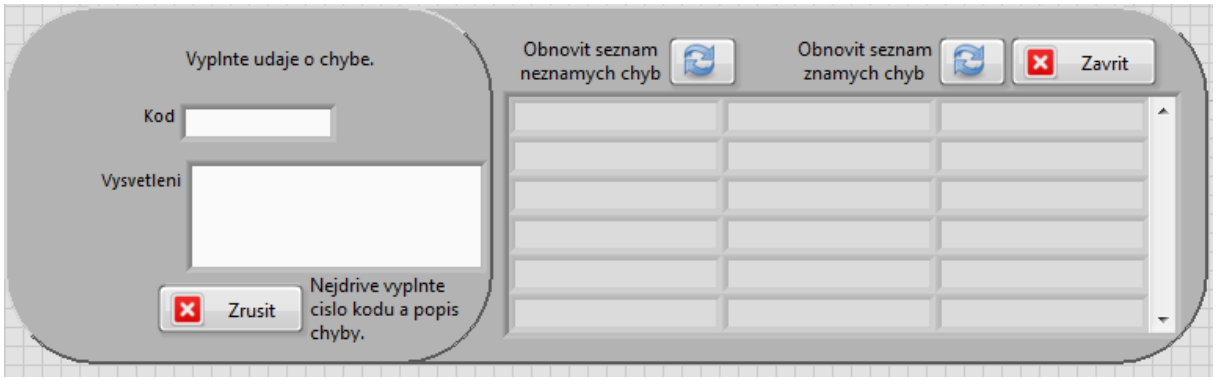


Obrázek 6.4: Čelní panel VI KnownErrors.vi prvního konceptu

Neznámé chyby je možné vysvětlit v podprogramu UnknownErrors.vi (Obrázek 6.5Obrázek 6.5) pomocí zadání příslušného kódu a nového popisu chyby. Podprogram UnknownErrors.vi umožnuje neznámou chybu identifikovat a po stisku tlačítka Obnovit seznam známých chyb se tato chyba v další iteraci smyčky WHILE aplikace Menu.vi už jeví jako chyba popsaná. Při jednom spuštění tohoto podprogramu lze vysvětlit neomezené množství chyb, tyto vysvětlené chyby se uloží do souboru Chyby.txt umístěného v adresáři aplikace, aby byly i při dalším spuštění aplikace k dispozici.

Výhodou tohoto řešení je možnost rychlého odladění aplikace, kvůli nízkému stupni složitosti, další výhodou je možnost vysledování a opravení případných chyb bez zastavení aplikace.

Mezi nevýhody tohoto provedení patří zapsání naměřených hodnot až na konci programu, možnost zastavení aplikace při nenavázání komunikace, odezva aplikace až za dobu stanovenou v periodě měření a velké zapoudření celé aplikace, což snižuje přehlednost.

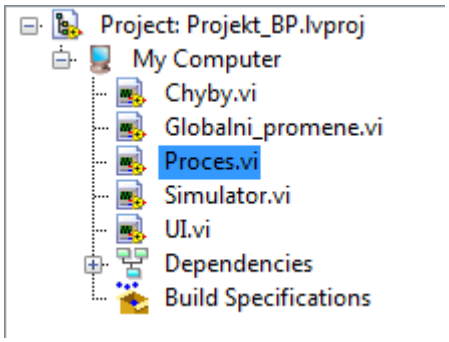


Obrázek 6.5: Čelní panel VI UnknownErrors.vi prvního konceptu

První koncept byl tedy po odzkoušení, odladění a konzultaci zamítnut.

6.3 Realizace druhého konceptu

Druhý koncept je vytvořen jako LabVIEW projekt (Obrázek 6.6), obsahuje jednu aplikaci, v této aplikaci obsahuje čtyři samostatné programy – Chyby.vi, Proces.vi, UI.vi a AY1020.vi, komunikující spolu přes globální proměnné. Souhrn globálních proměnných je rovněž součástí projektu. Tento koncept představuje realizaci preemptivního multitaskingu. Ten vytváří dojem paralelismu aplikace.



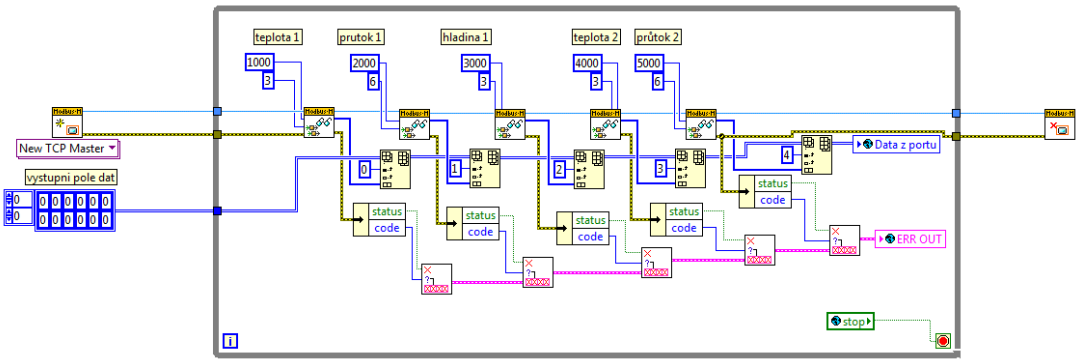
Obrázek 6.6: Projekt druhého konceptu

Program Chyby.vi se stará o řešení chybových stavů. Chyby zde vstupují z měřícího procesu (ERR COMM) a vyhodnocovacího procesu (ERR PROC), další částí je vizuální část skládající se z podprogramu KnownErrors.vi a UnknownErrors.vi popsaných v prvním konceptu. Program Chyby.vi po ukončení WHILE smyčky zapíše aktuální chyby v procesu do souboru s časovým razítkem v názvu.

Další části tohoto konceptuálního řešení je program Proces.vi, které má na starosti cyklické vyhodnocování získaných PDI zpráv a jejich přípravu pro vizualizaci a zápis. Po ukončení WHILE smyčky programu Proces.vi jsou zapsány připravené data pro zápis. Soubor se zapsanými daty je opatřen časovým razítkem v názvu pro lepší identifikaci při častějším měření.

Poslední částí druhého konceptu je vlastní měřící program s názvem Mereni.vi (Obrázek 6.7). V tomto programu dochází k cyklickému čtení dat z adres převodníku rozhraní AY1020. Po spuštění

programu se naváže komunikace s převodníkem rozhraní AY1020, po ukončení smyčky WHILE se přeruší komunikace.



Obrázek 6.7: Blokový diagram VI Mereni.vi druhého konceptu

Výhodou tohoto konceptu je samostatnost jednotlivých programů. Velkou nevýhodu představuje hrozba přepsání globální proměnné ze dvou míst najednou. Poté by se mohlo například stát, že by se část aplikace ukončila, zatímco zbytek by stále běžel. Tím by se v některých případech zamezilo ukončení zbytku aplikace a tím by se způsobilo zacyklení celého programu. Další nevýhoda je stejná jako u prvního konceptu a tou je dlouhá doba reakce.

Druhý koncept byl rovněž po realizaci zamítnut.

6.4 Realizace třetího konceptu

Třetí koncept představuje nejpokročilejší práci kvůli použité architektuře, tzv. Message Structure (Zprávové struktuře), která se používá při vývoji aplikací i v průmyslové praxi (Příloha 3). Tato architektura využívá systému front a notifikátorů pro předávání dat mezi paralelně běžícími smyčkami a tím umožní zrychlení běhu programu a také fyzické rozdělení logických částí programu. Další výhodou tohoto konceptu je skutečnost, že kvůli rozšíření tohoto principu vytváření aplikací, je snadné přenést práci na jiného vývojáře.

Třetí koncept byl vybrán jako finální verze programu.

Koncept se skládá ze smyčky čekající na aktivitu na čelním panelu, smyčky představující realizaci hlavního stavového automatu, dvou smyček představují vedlejší stavové automaty (měřící a logovací smyčky) a vizualizace naměřených dat.

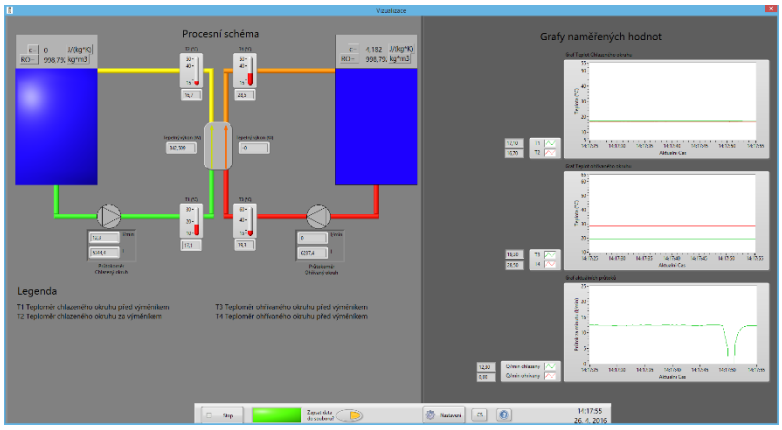
Vizualizační okno zobrazuje aktuální procesní hodnoty, graf průběhu teplot chlazeného okruhu, ohřívaného okruhu a také graf aktuálních průtoků za minutu. Všechny tyto grafy jsou s historií, což znamená, že umožňují ukázat půlminutový průběh. V spodní části vizualizace se nachází nástrojová lišta s tlačítky a aktuálním ukazatelem data a času (Obrázek 6.8).

Vstupem do vizualizační smyčky je fronta s daty pro grafy a notifikátor pro ostatní grafické prvky zobrazující aktuální hodnoty. Důvod rozdělení těchto dat do fronty a notifikátoru je nutnost zajištění všech dat v grafu tak, pro zobrazení celistvého průběhu.

Rozsahy vizualizačních grafických prvků i grafů jsou pevně nastaveny a neumožňují tzv. AutoScale (automatické změny rozsahů pro zvýšení přesnosti vizualizace). Vizualizace má také, kromě chráněné části, dvoujazyčné provedení (čeština a angličtina), což umožnuje provádět měření i zahraničním studentům.

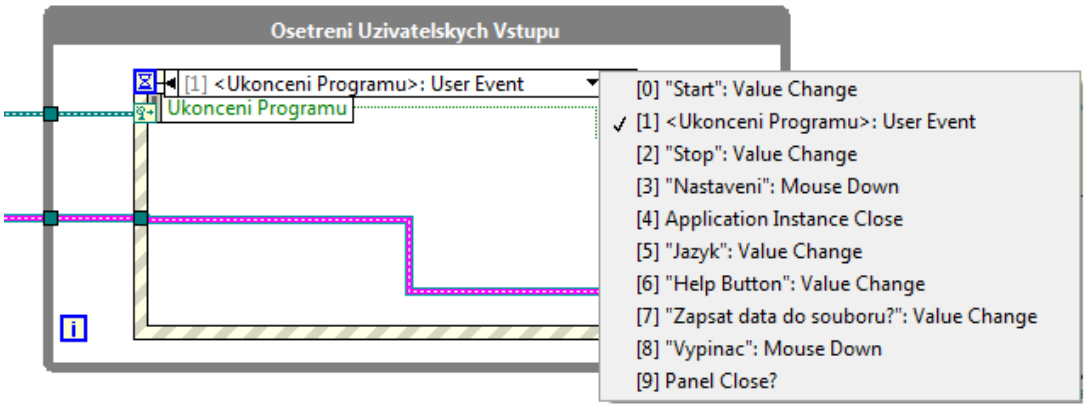
Aplikace má i nastavovací sekci, ve které lze nastavit adresu převodníku rozhraní AY1020 a umístění adresáře pro soubory s daty. Nastavovací dialog je přístupný po stisku tlačítka Nastavení na nástrojové liště programu. Logika sekce s nastavením umožňuje správci po poruše načíst poslední správnou konfiguraci, nebo také nově vytvořenou konfiguraci při ukončení aplikace uložit. Nastavovací sekce je chráněná před zásahy studentů přihlašovacím dialogem, proto tato část není dvoujazyčná.

Na nástrojové liště je ještě kromě tlačítek pro obsluhu měření, logování, ukončení aplikace a nastavení také tlačítko pro spuštění dialogu nápovědy. Po stisknutí tohoto tlačítka se otevře internetový prohlížeč s vytvořenou nápovědou k této aplikaci. Nápověda bude popsána níže spolu se souborem ReadMe.



Obrázek 6.8: Vizualizace finální aplikace za běhu

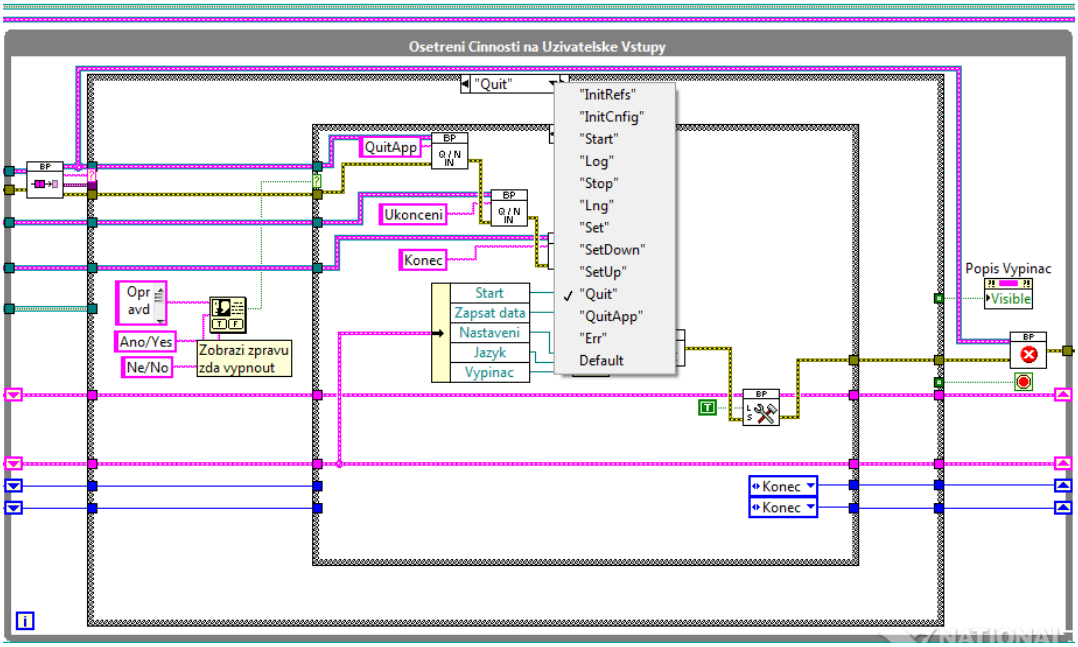
Smyčka čekající na aktivitu na čelním panelu řeší události (Obrázek 6.9), jako například stisk tlačítek, ukončení aplikace či zavření okna aplikace. Řešení událostí spočívá v ukládání příslušné zprávy do zprávové fronty (UI fronty).



Obrázek 6.9: smyčka Osetreni Uzivatelskych Vstupu třetího konceptu

Smyčka hlavního stavového automatu představuje hlavní část celé aplikace (Obrázek 6.10). Zde se příchozí zprávy (ze smyčky Osetreni Uzivatelskych Vstupu) vybírají z UI fronty a provádí se požadované činnosti, jako zapnutí a vypnutí měření, zpřístupnění tlačítek, změna jazyku aplikace,

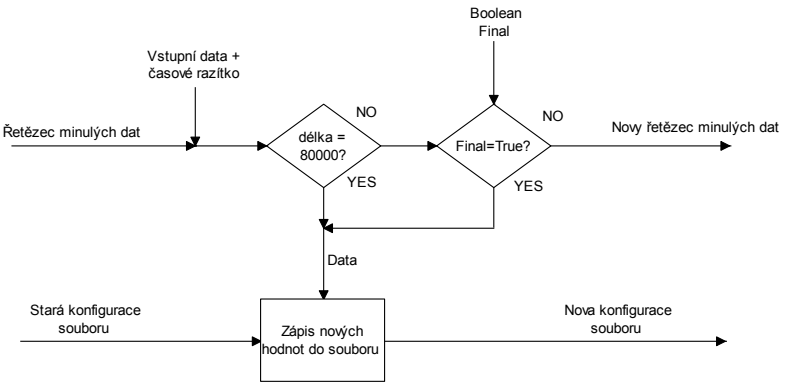
spuštění dialogu aplikace a podobně. Na konci každé iterace se vyhodnotí případná chyba. Pokud se chyba v procesu vyskytla, zapíše se do fronty zpráva Err, tato chyba se zobrazí a program se ukončí.



Obrázek 6.10: smyčka Osetreni Cinnosti na Uzivatelske Vstupy třetího konceptu

Další část aplikace je tvořena vedlejším stavovým automatem - logovací smyčkou, která se stará o správu souboru s logem. Práce se souborem je rozdělena na tři části. Při spuštění aplikace se vytvoří soubor Mereni – aktualni_cas.csv. Formát souboru je Comma Separated Value, tento formát umožňuje zapsaná data zobrazit v programu Microsoft Excel, a pokud jsou tyto data oddělena středníkem, Excel je umí rozdělit do buněk. Takové řešení umožnuje snadno zpracovat naměřená data do grafů a tabulek. K vytvořenému souboru je připsána hlavička o celkové délce řetězce 102 znaků.

Při stavu Log (Zápis do souboru) se poté naměřená data připravují na zápis přepisem na textový rětězec o délce přibližně 80 znaků. Připravená data se zapíšou až po dosažení 1000 iterací stavu Log, kdy dosahuje celková délka řetězce přibližně 80000 znaků (Obrázek 6.11). V případě že není logování aktivní, posílají se dosud nezapsaná data do další iterace.



Obrázek 6.11: Rozhodovací logika zápisu dat

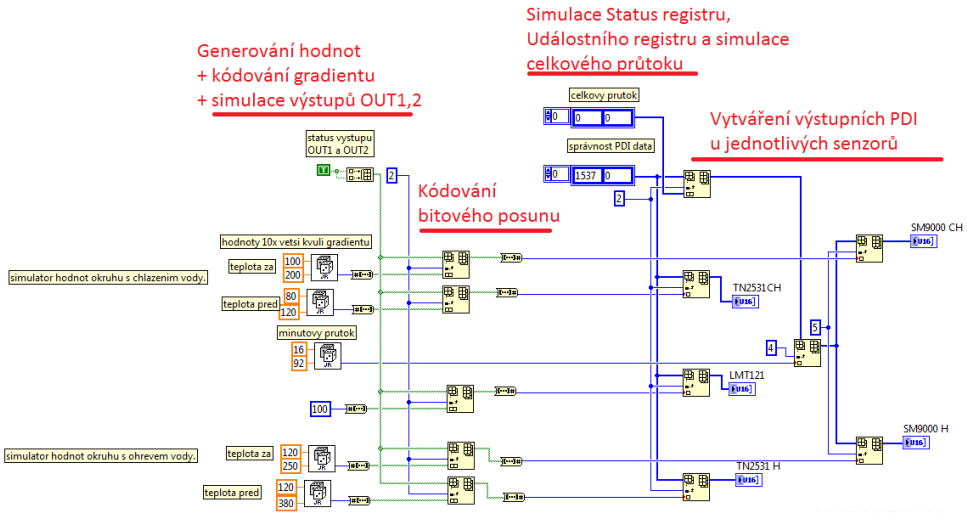
Při ukončení aplikace se data, která se nezapsala, zapíšou bez ohledu na délku připraveného řetězce. Pokud se do souboru nezapsaly žádné data kromě hlavičky a jednoho prázdného řádku, uvedený soubor se po ukončení smaže. Při změně složky souboru v nastavovací části se již vytvořený soubor přesune do nové složky.

Druhý vedlejší stavový automat řeší Měřící a vyhodnocovací smyčka. Komunikace s převodníkem rozhraní AY1020 se inicializuje po stisku tlačítka Start a ukončí se stiskem tlačítka Stop.

V této části se periodicky vyčítají data z Holding registrů převodníku rozhraní AY1020, tyto data se vyhodnotí, provedou se požadované výpočty a naměřená data se zapíšou do příslušných front a notifikátoru.

V případě změny parametrů komunikace se provede tzv. Teplý restart. Ten spočívá v ukončení stávající komunikace a inicializování nové komunikace s novými parametry.

V době, kdy nebylo pracoviště připravené pro testování vytvořených aplikací, nebo při domácím vývoji, kde dlouho probíhal majoritní vývoj aplikace, bylo nutné nahradit reálnou komunikaci simulátorem procesních hodnot pracoviště (Obrázek 6.12).



Obrázek 6.12: Blokový diagram SubVI Simulator.vi

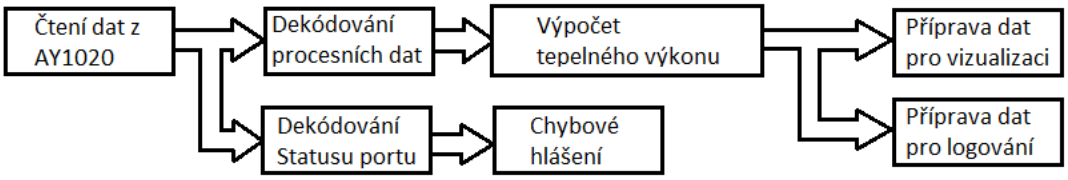
Simulátor má čelní panel uzpůsoben pouze pro potřeby konektoru. Simulátor funguje na principu generování náhodných hodnot (Tabulka 6.1) v určitých rozsazích, kódování těchto hodnot a následném přidávání do polí pro vytvoření zprávy podobné PDI z převodníku rozhraní.

Tabulka 6.1: Rozmezí generování hodnot, CH = chlazený okruh, H = Ohřívaný okruh

Typ senzoru	rozsah generování hodnot	jednotka
LMT121	100	%
SM9000	9,2-10,8	l/min
	20-30 CH, 25-32 H	°C
TN2531	12-20 CH, 38-50 H	°C

Na reálném pracovišti je nutné číst data z komunikačního protokolu Modbus. Použití Modbus funkcí pro čtení parametrů z převodníku rozhraní AY1020 bylo vysvětleno u realizace druhého konceptu (Obrázek 6.7) a v příslušné teoretické kapitole (Kapitola 3.2.3).

Cyklicky volané zpracování dat (Příloha 2 a Obrázek 6.13) se skládá z čtení dat z Modbusu, dekódování čtených zprav, výpočtu tepelného výkonu a přípravy dat pro vizualizaci a zápis do souboru.



Obrázek 6.13: Proces zpracování čtených dat

Po navázání komunikace se data vyčítají cyklicky z adres převodníku rozhraní AY1020 dle následující tabulky (Tabulka 6.2). Po zisku dat je nutné data dekódovat. Pro dekódování těchto dat (PDI) byly vytvořeny 3 tzv. dekódovací programy, každý pro jeden typ připojeného senzoru.

Dekódovací programy vychází ze společného principu a liší se pouze dekódováním různého objemu procesních dat, proto je princip dekodéru popsán na nejsložitějším programu, který dekóduje PDI z senzoru SM9000 – zařízení P1T2 a P2T4 (Obrázek 6.14). Ostatní blokové diagramy dekódovacích programů jsou uvedeny v přílohách, neboť se mírně liší.

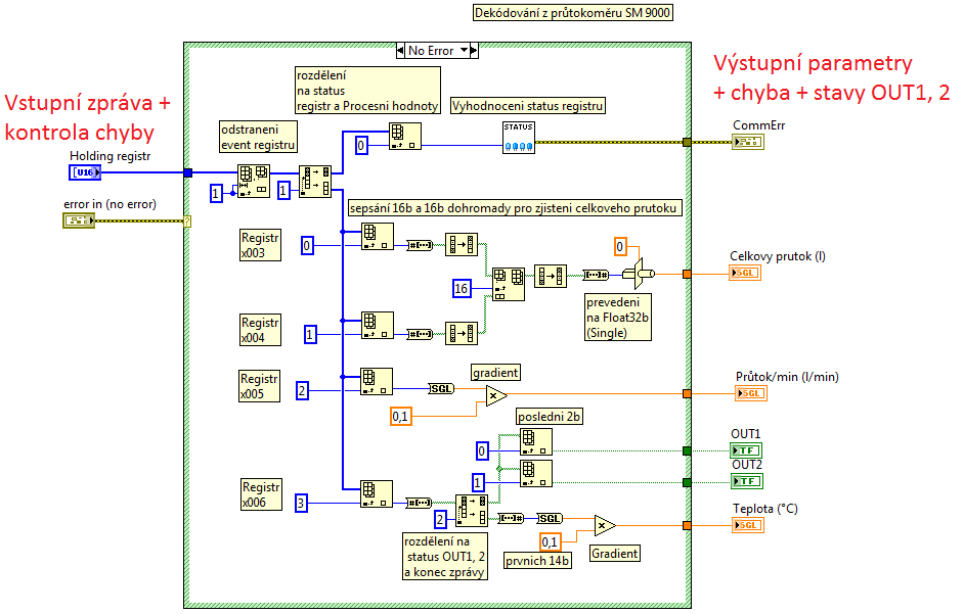
Tabulka 6.2: Konfigurace parametrů pro čtení z převodníku rozhraní AY1020

Zařízení	připojený port převodníku AY1020	Počáteční adresa	Koncová adresa
T1	1	1000	1003
P1T2	2	2000	2006
T3	4	4000	4003
P2T4	5	5000	5006
L1	3	3000	3003

Vstupní zpráva se rozdělí na jednotlivé registry, naměřené data se poté dle příslušné kódovací tabulky (Tabulka 4.1 až Tabulka 4.3) dekódují a zobrazí. Současně se s dekódováním naměřených dat vyhodnotí stav připojeného portu a stav spínaných výstupů OUT1 a OUT2.

V dekódovacích programech se nachází podprogram Status.vi. Status.vi vyhodnocuje funkčnost portu dle stavů jednotlivých bitů v status Bytu (str. 31). Podle stavu portu se na výstup přiřadí odpovídající chybový kód (kódy 5000-5004 v Tabulka 6.4).

Pro vyhodnocení všech parametrů byl vytvořen ještě program Vykon.vi. Ten počítá ze vstupních parametrů (Teplota před výměníkem, Teplota za výměníkem a aktuální minutový průtok) předaný výkon mezi okruhy ve Wattech (6.1):



Obrázek 6.14: Blokový diagram podprogramu DecodeSM9000

$$P = \frac{Q_v}{60000} \cdot \rho \cdot c \cdot \Delta t \cdot 1000 \qquad (6.1)$$

Kde: P je tepelný výkon (W)

c je měrná tepelná kapacita ($\frac{J}{kg \cdot K}$) pro vodu je to hodnota 4,182 $\frac{kJ}{kg \cdot K}$

Δt je rozdíl teplot na vstupu a výstupu (t_1-t_2) na jedné straně výměníku (°C)

Q_v je objemový tok (m3/s)

60000 představuje přepočet objemového průtoku z l/min na m^3/s

ρ je hustota (kg/m^3), pro vodu je tato hodnota přibližně 1000 kg/m^3

Hodnota hustoty je teplotně závislá [7], tato teplotní závislost byla po zjednodušení zahrnuta při výpočtu tepelného výkonu (Tabulka 6.3) a rovněž byla zobrazena.

Tabulka 6.3: Teplotní závislost hustoty v rozmezí teplot 4-40 stupňů

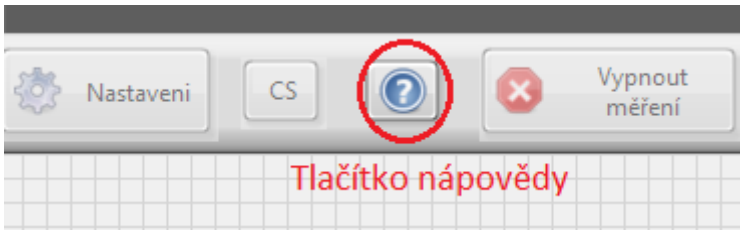
Rozmezí teplot (°C)	Hodnota hustoty (kg/m^3)
4-	999,973
4-10	999,701
10-20	998,205
20-30	995,651
30-40	992,220
40+	988,040

Třetí koncept definuje celkem 5 uživatelské kódů chyb (Tabulka 6.4). Chyby jsou definované v komunikačním procesu a další vytvořená chyba ošetřuje špatnou zprávu ve zprávové struktuře.

Tabulka 6.4: Popis definovaných Error kódů

Kód chyby	Chybová událost
5000	Chybí informace o stavu portu
5001	Inicializace IO-Link portu je aktivní
5002	Připojený senzor je provozuschopný, ale neposílá data
5003	Vyskytla se opravitelná HW či SW chyba na senzoru či portu
5004	Vyskytla se neopravitelná HW či SW chyba na senzoru či portu
5005	V zprávové struktuře se vyskytla neočekávaná zpráva

Pro třetí koncept byly vytvořeny dva soubory s HTML syntaxí a to ReadMe a nápovědu. ReadMe soubor obsahuje informace o verzi sestavené aplikace, minimální HW konfiguraci, nutné verzi LabVIEW a vývojářích aplikace. Soubor nápovědy obsahuje základní popis aplikace, vzorce s výpočty, doporučení a upozornění na určité kritické sekce. Dialog nápovědy je možné z aplikace spustit kliknutím na tlačítko Help, která má vizuální podobu otazníku (Obrázek 6.15). Nápověda je vytvořena ve dvou jazycích, češtině a angličtině.



Obrázek 6.15: Část čelního panelu s tlačítkem nápovědy

Nápovědu lze kontextově rozdělit do čtyř částí. V první části je stručně popsán proces měření, s ním spojené prvky čelního panelu a výpočet tepelného výkonu. V druhé části je popsán proces

logování a s ním související prvky čelního panelu, v třetí části je popsána vizualizace procesních hodnot a v poslední části je vysvětlen nastavovací dialog (Obrázek 6.16).

UI.vi

Requires: Base Package, Modbus Library

Platform(s): Windows

Aplikace **Mereni vymeny tepla** kontinualne **meri**, **vyhodnocuje**, **zobrazuje** a **loguje** vybrana data do souboru. Tato aplikace byla vytvorena v ramci Bakalarske prace a bude slouzit studentum oboru Merici a ridici systemy.

Application **Heat Exchange Measurement** is continously **Measuring**, **Calculating**, **Showing** and **Logging** choosed data into file. This Application has been developed as a part of Bachelor Thesis and will be useful for students studying Measure and Control systems .

Tato napoveda obsahuje tyto casti:
This help dialog consist of these parts:

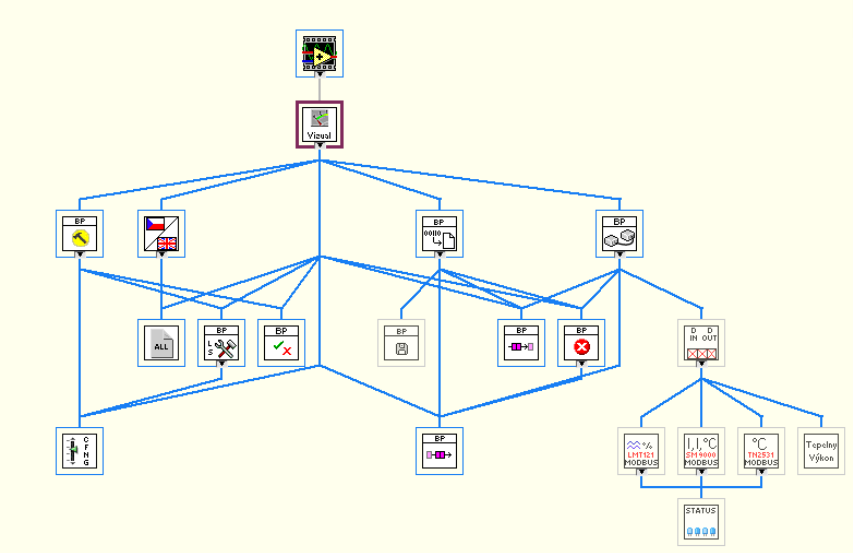
Mereni/Measurement

Logovani/Logging

Vizualizace/Visualization

Nastavovaci dialog/Setting dialog

Obrázek 6.16: Úvodní část nápovědy



Obrázek 6.17: VI hierarchie třetího konceptu

Koncept byl po schválení odladěn a sestaven do projektu. Projekt obsahuje šestnáct podprogramů, dvě nové definice datových typů, čtyři podpůrné soubory a ikonu (Obrázek 6.17). Z projektu byla vytvořena spustitelná aplikace, která byla poskytnuta k testování a doladěna. Spustitelná aplikace byla umístěna na plochu připojeného počítače, ostatní soubory byly umístěny do adresáře C:\MereniVymenyTepla, tak aby se zamezilo zásahům studentů do projektu.

6.5 Testování

Spustitelná aplikace byla poskytnuta studentům pro ověření funkčnosti a dostatečné ochraně chráněných sekcí. Testoval se kromě SW aplikace i HW model, konkrétně rychlost chlazení chladičem a topení topnou spirálou a také teplotní spád výměníku.

6.5.1 Odladění aplikace

V průběhu testování sestavené aplikace (Vymena_Tepla.exe) se odladily následující detaily:

- Úprava V1.0.0.0 – Vytvoření hotové aplikace
- Úprava V1.0.1.0 - Upravilo se umístění doprovodných souborů obsahujících přístupové ID a hesla (soubor ../Security.txt) a dostupné konfigurace (soubor ../Config.txt).
- Úprava V1.0.2.0 – Odladila se česká verze nápovědy – bylo nutné odstranit veškerou diakritiku.
- Úprava V1.0.3.0 – Aplikaci se přidělala ikona a otestoval se dialog nápovědy
- Úprava V1.0.4.0 – Úprava horních a dolních mezí zobrazování hodnot + oprava historie grafu (upraveno z 1024 vzorků na 540, což odpovídá 30 sekundám) + oprava zobrazování aktuálních hodnot průtoku – přehozené průtokoměry

6.5.2 Testování funkčnosti přípravku

Pracoviště bylo podrobeno sérii testů. Měřily se tyto průběhy:

- Rychlost ochlazení média v chlazené nádrži z teploty 25 °C na teplotu 15 °C, při dvou rychlostech oběhového čerpadla pro zjištění vlivu ohřívání oběhového čerpadla na chladící proces
- Rychlost ohřevu média v ohřívané nádrži z teploty 25 °C na teplotu 40 °C, pro poloviční a plný výkon topné spirály
- Rychlost teplotního spádu pro dvě rychlosti oběhových čerpadel pro získání průběhů teplot v okruzích

Čerpadlo pro účely testování pracuje na nízké rychlosti (cca 12,5 l/min) a vysoké rychlosti (cca 19 l/min).

Při testování rychlosti ochlazení média byl systém v následujícím stavu:

- Počáteční teplota média v okruhu: 25 °C
- Koncová teplota média v okruhu: 15 °C
- Zapnuté oběhové čerpadlo chlazeného okruhu, čerpadlo chladiče a chladič
- Vypnutá topná spirála a oběhové čerpadlo ohřívaného okruhu

Výsledkem měření jsou charakteristiky (Příloha 6), které ukázaly, že použitý chladič ochladil příslušnou nádrž o 10 °C za více než hodinu při nízké rychlosti čerpadla. Při vysoké rychlosti čerpadla se plně projevil efekt chlazení čerpadla do okruhu. Tímto se zvýšil čas chlazení přibližně o další hodinu (viz. obrázky v příloze 6). Bohužel jsou tyto časy příliš velké, a proto je použitý chladič nevhodný pro potřeby laboratorního měření.

Při testování rychlosti ohřevu teplonosného média v okruhu byl systém v následujícím stavu:

- Počáteční teplota média v okruhu: 25 °C
- Koncová teplota média v okruhu: 40 °C
- Zapnuté oběhové čerpadlo ohřívaného okruhu a topná spirála
- Vypnuté oběhové čerpadlo chlazeného okruhu, čerpadlo chladiče a chladič

Výsledkem měření jsou dvě charakteristiky rychlosti ohřevu ohřívané nádrže zobrazené v přílohách (Příloha 5). Charakteristiky ukazují rychlost ohřevu média v ohřívané nádrži měření, jehož výsledkem je porovnání rychlosti ohřevu při polovičním výkonu (1kW) a plném výkonu (2kW) topné

spirály. Rychlost ohřevu média při plném výkonu spirály (cca 30 minut) je použitelná pro provedení laboratorního měření. Toto měření poukázalo rovněž na skutečnost, že topná spirála není schopná vytopit nádrž na více než 51 °C. Je to kvůli odkrytí a nezaizolování nádrže, což způsobuje ztráty.

Při testování rychlosti teplotního spádu byl systém v následujícím stavu:

- Počáteční teplota chlazeného média: 15 °C
- Počáteční teplota ohřívaného média: 40 °C
- Zapnuté oběhové čerpadla procesních okruhů, vypnuté čerpadlo chladiče, chladič a topná spirála

Výsledkem měření jsou charakteristiky průběhu teplotního spádu pro různé stupně rychlostí čerpadel. Při měření se zjistilo, že použitý výměník tepla je pro tuto aplikaci dostačující, byť by jeho správné - protiproudé zapojení zvýšilo efektivitu přenosu tepla. Nicméně i tak je výměník schopný dosáhnout výkonu až 12 kW při výměně tepla s jiným okruhem. Teplotní spád 25 °C je tedy schopný vyrovnat za přibližně 25 minut (Příloha 4).

Závěr

V této práci jsem zautomatizoval měření laboratorní úlohy předání tepla dle zadání práce. Realizace práce probíhala po částech. Nejdříve jsem si všechny potřebné komponenty elektricky připojil a zajistil nutnou komunikaci mezi senzory a počítačem.

V další části jsem si určil parametry výsledné aplikace a dle nich jsem postupně navrhl tři koncepty. Výsledná aplikace měla zajišťovat periodické měření procesních hodnot, jejich následné zpracování (převedení, potřebné výpočty apod.), dle rozhodnutí uživatele také zápis zpracovaných hodnot do souboru. Aplikace měla mít dle zmíněných parametrů rovněž odpovídající vizualizační okno.

Třetí částí mé práce představovala tvorbu kódů pro realizaci konceptů. Tyto koncepty byly následně porovnány. První dva koncepty byly pro nedostatečné ošetření možných chyb vedoucím práce odmítnuty, realizace třetího konceptu vyústila ve výsledné řešení.

Výsledkem mé práce je tedy aplikace s komplexní zprávovou strukturou imitující řešení stavového automatu s funkčností požadovanou při návrhu konceptů. Aplikace má rovněž množství rozšiřujících parametrů.

Mezi tyto rozšiřující parametry patří umožnění oprávněným osobám změnit základní parametr komunikace (IP adresu převodníku rozhraní AY1020) a také změnit umístění souboru, do kterého jsou zapsány naměřené hodnoty. Další doplňkovou vlastností je zobrazování přesnější hodnoty hustoty pro aktuální hodnotu teploty, a také možná změna jazykového provedení vizualizace aplikace mezi češtinou a angličtinou.

Pro tuto aplikaci byl rovněž vytvořen soubor ReadMe.html obsahující stručné informace pro dalšího vývojáře aplikace o aktuální verzi, požadavcích na systém a dalších užitečných informacích. Druhý vytvořený doprovodný soubor je nápověda UI.html. Tento soubor lze otevřít kliknutím na tlačítko Help ve vizualizaci. Soubor nápovědy je rovněž dvoujazyčný a obsahuje vysvětlení funkčnosti aplikace pro běžného uživatele.

Při testování parametrů pracoviště jsem zjistil, že by bylo vhodné vyměnit chladič za silnější, neboť pro ochlazení nádrže potřebuje dobu delší než je délka vyučování. Naopak plný výkon topné spirály je dostačující.

I když je výměník zapojen v souproudém provedení, lze na něm demonstrovat účinnost deskových výměníků, neboť dokáže srovnat teploty v obou nádržích do půl hodiny dle momentálního výkonu čerpadel. Zapojení tohoto výměníku do protiproudého provedení by bylo nejspíše kontraproduktivní.

Provedení práce umožňuje další vývoj, který by se mohl zaměřit na doplnění aplikace o stykače či SSR relé pro umožnění řízení činnosti chladiče a topné spirály. Po těchto úpravách bude možné vytvořit až šest možných regulačních úloh, kde by si studenti mohli mimo jiné vyzkoušet manuální ovládání soustavy nebo návrh zpětnovazebního řízení.

Další možný vývoj lze zaměřit na úpravu zprávové struktury. Lze jí například zjednodušit a zprávy předávat v jedné zprávové frontě místo třech, čím se sníží paměťové nároky aplikace, ale zvýší nároky na logické myšlení vývojáře. Jiný vývoj zprávové sktruktury představuje upravení logovacího formátu na formát TDMS.

Neposlední možnost vývoje nabízí doplnění aplikace o logování všech vyskytnutých chyb, či ošetření procesních chybových stavů jako například čerpání z prázdné nádrže.

Byť nechci provést tento vývoj v rámci diplomové práce, velice rád bych byl u tohoto vývoje. Mít tak o půl roku více času a připravené doplňkové vybavení, provedl bych vývoj sám a začlenil jej do této bakalářské práce.

Bibliografie

[1] ROZSYPAL, Stěpán. *Výměníky tepla* [online]. Brno, 2010, 2010-05-27 [cit. 2016-01-04]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=29102. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Marek Balaš.

[2] Deskové výměníky Alfa Laval. *BCB Plzeň* [online]. Plzeň: BCB Plzeň, 2013, 2013-04-13 [cit. 2016-01-04]. Dostupné z: http://www.bcb-plzen.eu/alfalaval/pajene.htm

[3] KLEIN, Marcus. *Industrial Ethernet - Challenges and Drawbacks: Comparision of Modbus TCP/IP and Ethernet IP* [online]. Vídeň, 2011, 2011-05-12 [cit. 2016-01-04]. Dostupné z: https://www.auto.tuwien.ac.at/bib/pdf_TR/TR0154.pdf. Bakalářská práce. Technische Universität Wien. Vedoucí práce Ao. Univ. Prof. Dr. Wolfgang Kastner.

[4] *The Modbus Organization* [online]. Spojené státy americké: The Modbus Organization, 2003, 2016-01-04 [cit. 2016-01-04]. Dostupné z: http://modbus.org/

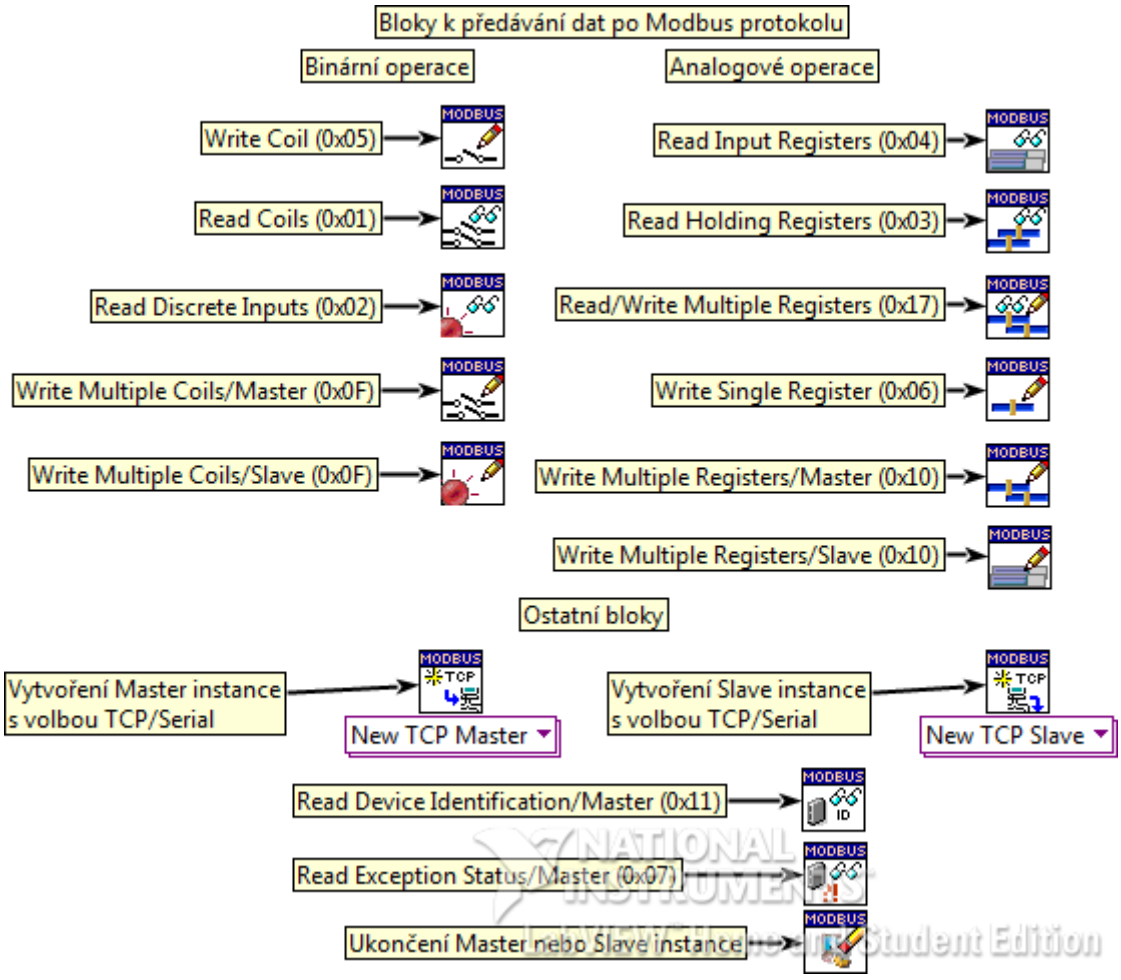
[5] VOJÁČEK, Antonín. IO-Link - popis digitální komunikace pro senzory. In: *Automatizace.hw.cz* [online]. 2010 [cit. 2016-04-16]. Dostupné z: http://automatizace.hw.cz/iolink-popis-digitalni-komunikace-pro-senzory

[6] *IFM electronics: sensors, networking and control systems* [online]. Průhonice: IFM electronics, b.r., 2016-01-04 [cit. 2016-01-04]. Dostupné z: http://www.ifm.com/ifmcz/web/home.htm

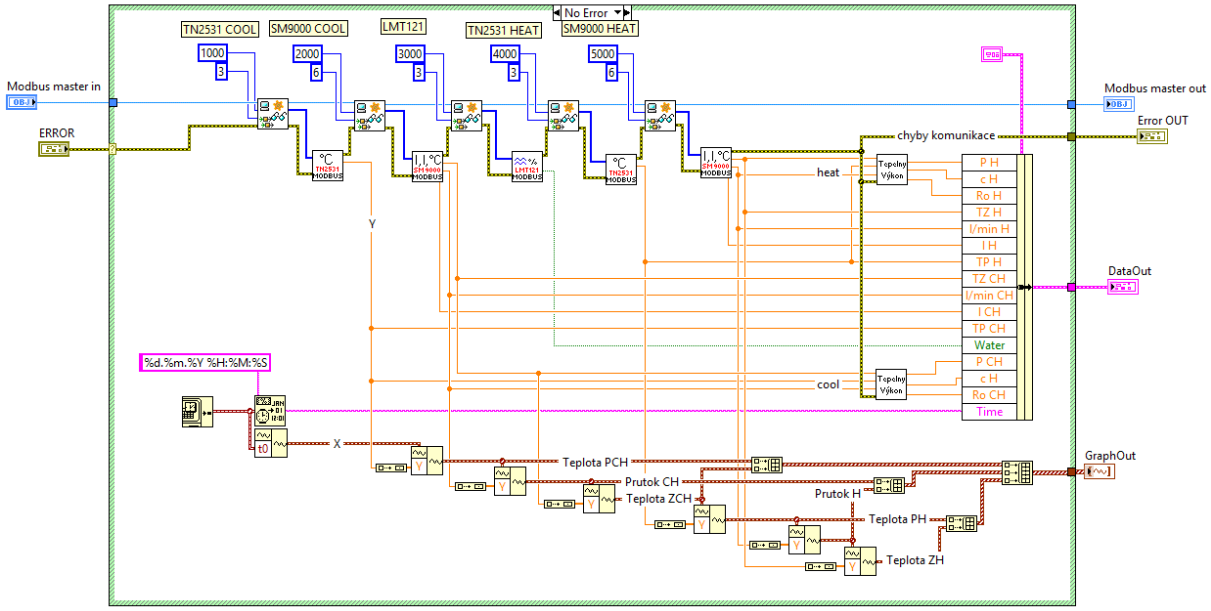
[7] Závislost hustoty destilované vody na teplotě. *ConVERTER: Převody jednotek, fyzikální tabulky, životopisy fyziků a Nobelova cena* [online]. Praha: Jiří Bureš, 2002 [cit. 2016-04-20]. Dostupné z: http://www.converter.cz/tabulky/hustota-vody.htm

Seznam příloh

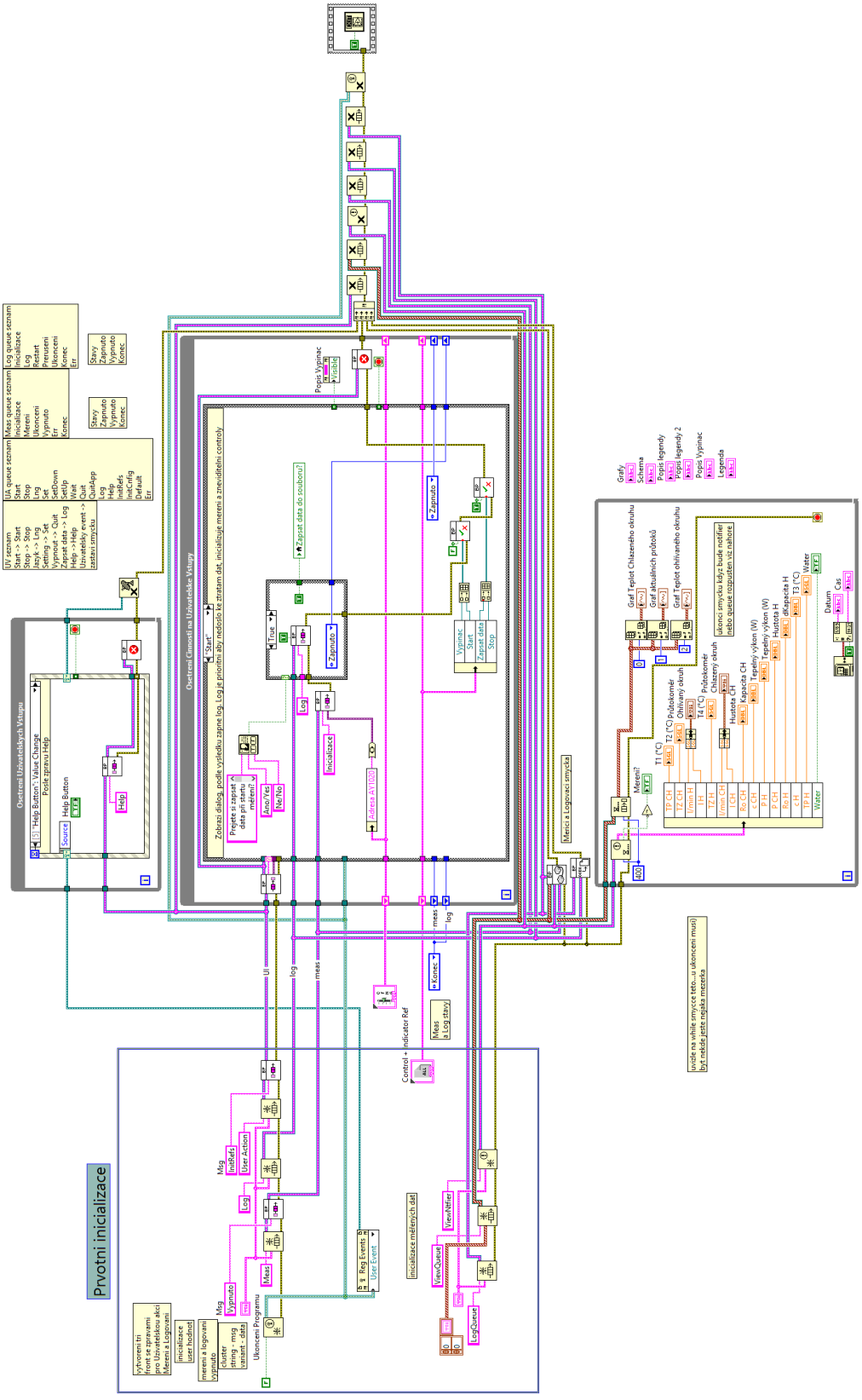
Příloha 1: LabVIEW funkce pro předávání dat v Modbus protokolu



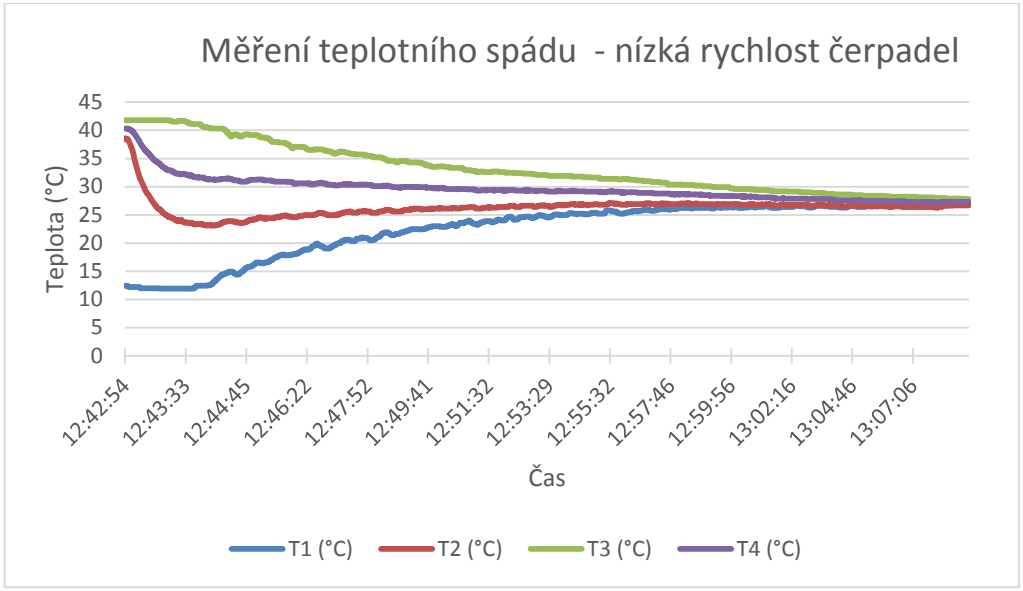
Příloha 2: Blokový diagram procesu měření a vyhodnocení a logovací logiky

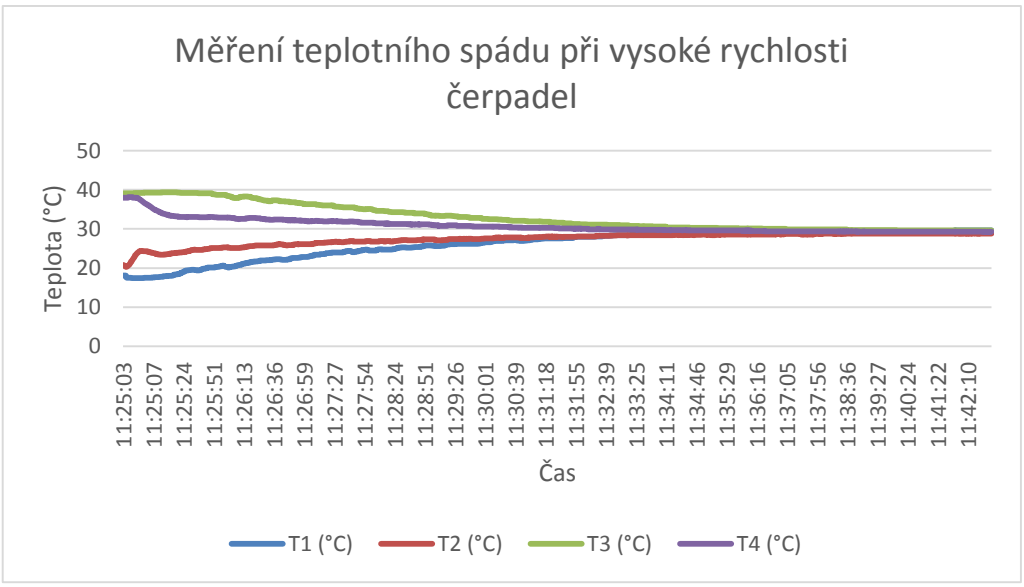


Příloha 3: Blokovy diagram finálního produktu

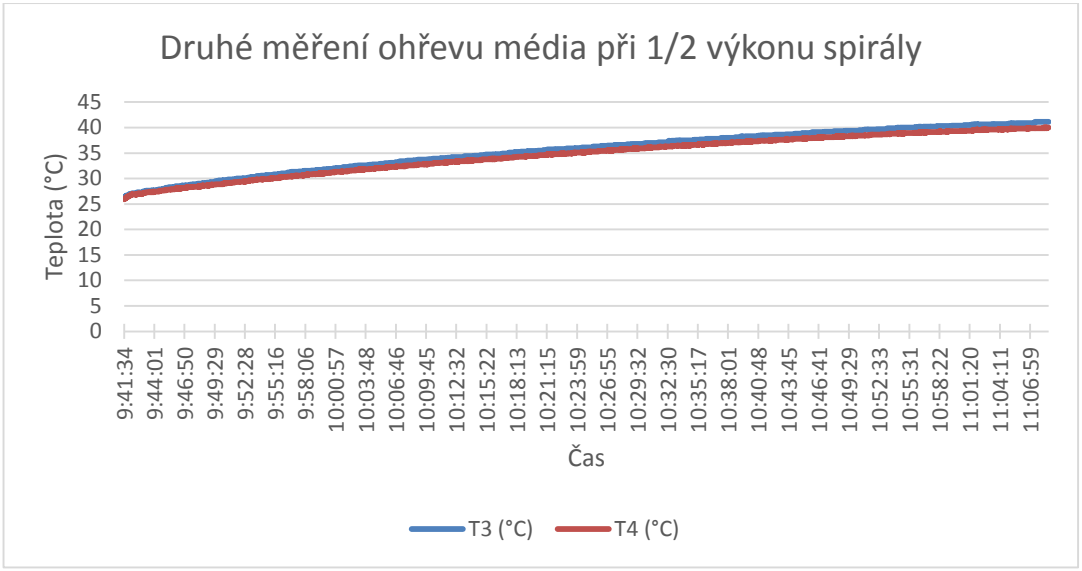


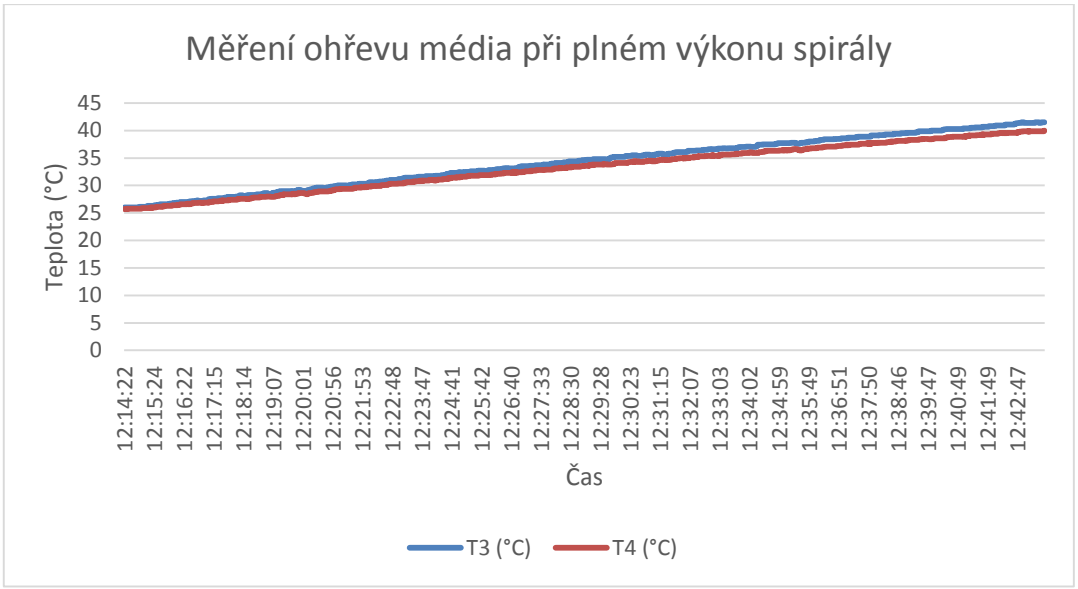
Příloha 4: Měření teplotního spádu při proměnné rychlosti čerpadla



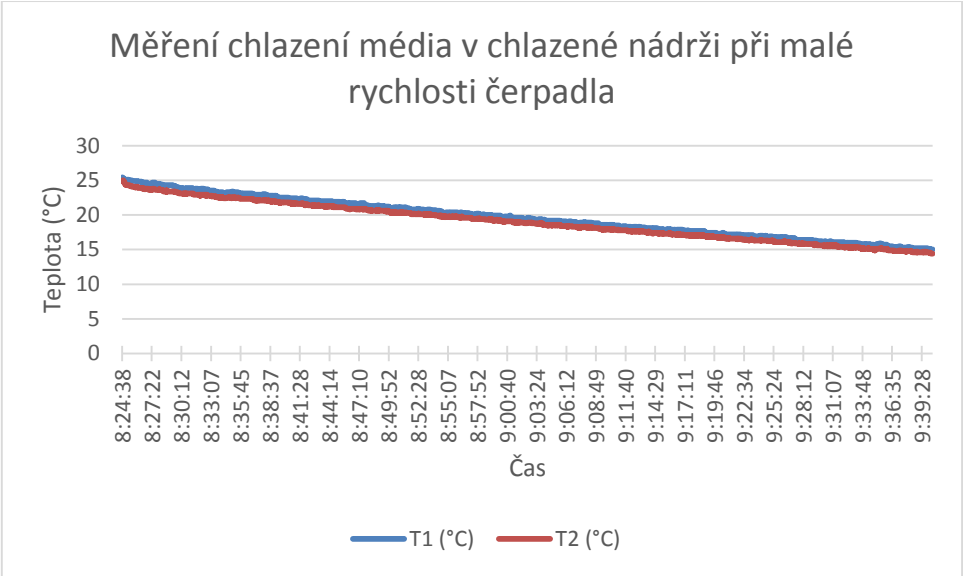


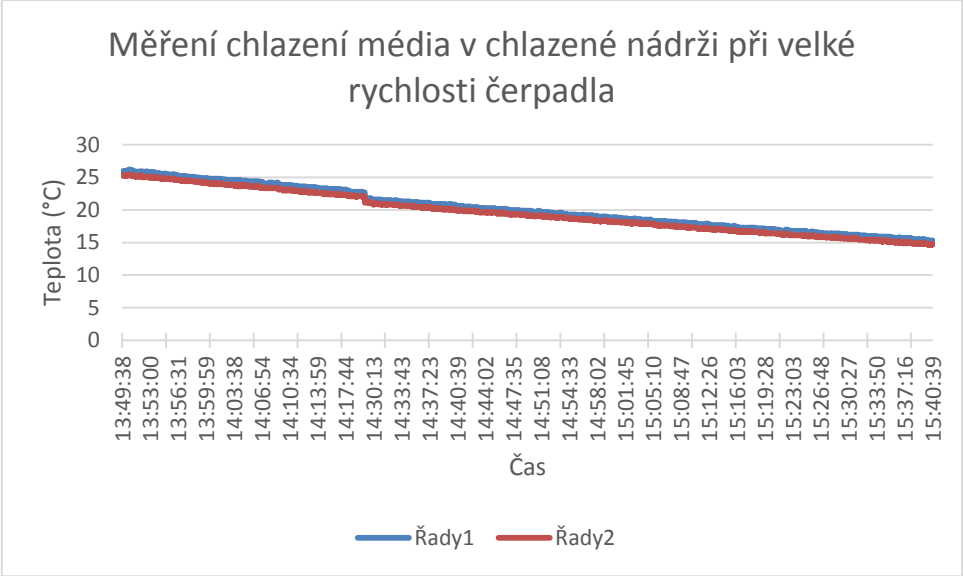
Příloha 5: Měření ohřevu média při proměnném výkonu topné spirály





Příloha 6: Měření chlazení média při proměnné rychlosti čerpadla





Příloha 7: Příloha na CD – Prilohy_BP

Obsah přiloženého CD:

- Soubory měřícího projektu (*.vi, soubory *.txt a soubory *.html, *.ico) v adresáři Projekt
- Sestavená aplikace
- Neupravený výstup měření teplotního spádu – soubor Mereni - 2016_04_27_11_10.csv